

Universidad de Alcalá
Escuela Politécnica Superior

GRADO EN INGENIERÍA INFORMÁTICA



ESCUELA POLITECNICA
Autor: Luis Ángel Parada Rodríguez
SUPERIOR
Tutor/es: Salvador Otón Tortosa

2017



Escuela Politécnica Superior

Grado en Ingeniería Informática

Trabajo Fin de Grado

Aplicación para Control Alimenticio

Luis Ángel Parada Rodríguez

Alcalá de Henares, Septiembre 2017

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

Grado en Ingeniería Informática

Trabajo Fin de Grado

Aplicación para Control Alimenticio

Autor: Luis Ángel Parada Rodríguez

Director: Salvador Otón Tortosa

TRIBUNAL:

Presidente:

Vocal 1º:

Vocal 2º:

CALIFICACIÓN:

FECHA:

INDICE

RESUMEN	1
PALABRAS CLAVE	1
ABSTRACT	3
KEYWORD.....	3
RESUMEN EXTENDIDO.....	5
INTRODUCCIÓN	7
1.1 INTRODUCCIÓN	7
1.2 OBJETIVOS DEL PROYECTO	8
1.3 ESTRUCTURA DE LA MEMORIA.....	8
MARCO TEÓRICO	12
2.1 TEMA 1: DESARROLLO WEB VS MÓVIL	12
2.1.1 DJANGO WEB FRAMEWORK.....	15
2.1.2 ELECCIÓN DE DJANGO	28
2.2. TEMA 2: AMAZON WEB SERVICES.....	39
2.2.1 AMAZON WEB SERVICES.....	39
2.2.2 COMPARATIVA AMAZON WEB SERVICES VS GOOGLE CLOUD VS MICROSOFT AZURE	49
2.3 TEMA 3: IMPORTANCIA DE UNA ALIMENTACIÓN BALANCEADA.....	54
2.3.1 OBESIDAD	54
2.3.2 ALIMENTACION BALANCEADA	59
MARCO PRÁCTICO.....	67
3.1. INTRODUCCIÓN.....	67
3.2 METODOLOGÍA.....	71
3.2.1 RECOGIDA DE REQUISITOS	71
3.2.2 CASOS DE USO	73
3.3 ARQUITECTURA DEL SISTEMA.....	79
3.4 IMPLEMENTACIÓN Y FUNCIONAMIENTO DEL SISTEMA.....	93
3.4.1 IMPLEMENTACIÓN.....	93
3.4.2 DESPLIEGUE A PRODUCCIÓN	97
CONCLUSIONES Y FUTURAS MEJORAS.....	103
PRESUPUESTO DEL PROYECTO.....	107
COSTE DE LA MANO DE OBRA	107
COSTE DEL MATERIAL	107
COSTE HARDWARE.....	107
COSTE SOFTWARE	108
GASTOS GENERALES	108
COSTE GLOBAL	108
BIBLIOGRAFÍA	111
MANUAL DE USUARIO.....	115
REGISTRO EN EL SISTEMA	115
AUTENTICACIÓN.....	123
APLICACIÓN PARA EL CLIENTE.....	128
MENU PRINCIPAL.....	128
DATOS PERSONALES.....	129
CITAS.....	130
REGISTRAR COMIDA.....	132
PROGRESO	135

APLICACIÓN PARA EL NUTRICIONISTA	136
MANTENIMIENTO.....	138
REGISTRAR CLIENTE.....	138
REGISTRAR ALIMENTO	139
APLICACIÓN PARA EL ADMINISTRADOR.....	140

RESUMEN

En este documento se explica el desarrollo de una aplicación que permita el seguimiento alimenticio de un paciente que se encuentre en un régimen nutricional, para así tener la mejor aproximación posible sobre la ingesta calórica del paciente, como también si el paciente es diabético poder mantener un diario de los niveles de glucosa. El objetivo de la aplicación es llevar un diario del paciente de su ingesta semanal para poder analizar su ingesta calórica y su curva de glucosa como también la capacidad de concertar citas con su doctor. La aplicación final será una aplicación web desarrollado en el Framework en Python DJANGO utilizando como backend una base de datos MySQL, todo esto publicado en la nube de Amazon Web Services.

PALABRAS CLAVE

eHealth, Alimentación, Amazon Web Services, Django.

ABSTRACT

This document explains the development of an application that allows the nutritional follow-up of a patient who is in a nutritional regime, in order to have the best possible approximation about the caloric intake of the patient, as well as if the patient is diabetic to be able to maintain a daily glucose levels. The purpose of the application is to keep a patient's diary of your weekly intake to be able to analyze your caloric intake and glucose curve as well as the ability to make appointments with your doctor. The final application will be a web application developed in the Python DJANGO Framework using as backend a MySQL database, all published in the Amazon Web Services cloud.

KEYWORD

eHealth, Feeding, Amazon Web Services, Django.

RESUMEN EXTENDIDO.

La Sociedad Española de Cirugía de la Obesidad y de las enfermedades metabólicas (SECO), en colaboración con otras ocho sociedades nacionales relacionadas con el diagnóstico, manejo y tratamiento de la obesidad en niños y adolescentes, han presentado esta mañana en Madrid un estudio pionero sobre la obesidad infantil en nuestro país titulado *“Initial approach to childhood obesity in Spain”*.

El estudio tiene como objetivo analizar los posibles tratamientos ante la incidencia de la obesidad infantil en Europa, y España en particular, que en los últimos años ha aumentado de manera alarmante.

El estudio se ha centrado en niños y adolescentes con edades comprendidas entre los **6 y los 18 años**.

En España, la obesidad ha aumentado del 7,4% al 17,0% en los últimos 25 años. Actualmente un 23% de la sociedad adulta española es obesa y un 38% sufre sobrepeso. **En niños y adolescentes el porcentaje sería de 25% (sobrepeso) y 15% (obesos).** Es decir, **“de cada 10 niños y adolescentes de 2 a 17 años, 2 tienen sobrepeso y 1 obesidad”**. Si atendemos a la obesidad mórbida, un 1,2% de la sociedad sufre esta patología (0,6% son hombres y el 1,8% mujeres) y se estima que este porcentaje podría mantenerse a estos niveles en los próximos años. Se considera obesidad grave cuando el Índice de Masa Corporal (IMC=peso en kilos dividido por talla en metros al cuadrado), es mayor de 35. El IMC idóneo, independientemente del sexo, sería entre 20 y 25 (en los niños se mide con otros parámetros –percentiles–).

Las causas de este incremento derivan de factores externos como, cambios sociales, estilos de vida, conductas familiares y malos hábitos alimenticios. Respecto al incremento de la denominada comida “basura” y la reciente polémica con las bebidas azucaradas, el *Dr. Vilallonga* aclara que “es el consumidor final el que decide lo que consume por lo que los padres o tutores tienen un papel relevante para guiar la alimentación de los menores”.

Aunque en la población española vemos un incremento de peso a través de los años, se puede ver como una nueva tendencia de estar en forma. Esto ha traído un nuevo sector económico que es el fitness que lo podemos ver desde los productos alimenticios, aplicaciones y redes sociales.

En la época en la que vivimos en la que los sistemas de Tecnologías de la Información y la Comunicación nos ofrecen grandes oportunidades en todos los campos de estudios surgen nuevas necesidades en las que los sistemas TIC son un gran apoyo para el crecimiento de nuevos estudios y la posibilidad de ampliación de dichos estudios e investigaciones. Este hecho también se ha trasladado al campo de la medicina donde han surgido nuevos sistemas para ayudar al trabajo de los profesionales. También se han implantado nuevos sistemas en la mejora de las relaciones entre pacientes y profesionales. Estos sistemas han dado paso a lo

que se ha denominado como eHealth. Estos sistemas permiten hacer un seguimiento, monitorización y tratamiento remotamente lo que facilita la comunicación entre las dos partes. Paralelamente al aumento anteriormente mencionado de la demanda de sistemas TIC, el mundo de los llamados smartphones desde su creación ha supuesto un gran avance en lo que al mundo de la informática se refiere. Nos permite estar conectados constantemente gracias a las redes sociales, podemos mantener una conversación en cualquier momento y en cualquier lugar. Esta conexión permanente brinda oportunidades a los desarrolladores de crear aplicaciones de gran ayuda en nuestro día. Aplicaciones de compra venta de productos, transportes compartidos, juegos, redes sociales... han conseguido hacerse un hueco en nuestro día a día.

Para un nutricionista convencional lo más común es indicar una dieta y esperar que la persona haga un diario a mano mes a mes de sus comidas, para luego analizarlo. A través de esta aplicación el usuario podrá de una manera más fácil registrar que come durante día a día y así su doctor poder analizarlo de una manera más rápida y sencilla. De esta manera se puede incentivar a que los pacientes al llevar una memoria de lo que se come así el doctor poder analizar estos datos y re formular un nuevo plan alimenticio

INTRODUCCIÓN

1.1 INTRODUCCIÓN

La aplicación que se describe en este documento pretende darle una nueva herramienta al nutricionista en nivel de análisis de la ingesta calórica, pudiendo ver que alimentos ha comido en cada específica comida con su gramaje exacto como también una estadística semanal de la ingesta calórica del paciente, por el lado del paciente brindarle una herramienta más sencilla y útil y precisa para poder mantener un diario de comidas como también poder concertar sus citas on-line.

Una encuesta presentada en un congreso de la Federación Española de Sociedades de Nutrición reveló que el **porcentaje de españoles que no se alimenta de** manera adecuada **asciende al 70%**. Por esto, recomiendan hacer ejercicio, evitar las harinas refinadas, las grasas y los azúcares.

Los malos hábitos alimenticios se reflejan en el hecho de que hoy **hay tres veces más de niños obesos que hace 15 años** y según datos de la Sociedad Española para el Estudio de la Obesidad (SEEDO), la población obesa que tiene entre 25 y 54 años es del 15,5%. La mayoría de los afectados por la pandemia mundial son mujeres.

Ya hay aplicaciones en el mercado que se sustentan en esta idea y tratan de resolverla con mecanismos similares, sin embargo la mayoría no operan en España, la problemática de esta aplicación es de donde sacar la información de la comida, la mayoría de este estilo e aplicaciones no usan datos oficiales y las que si lo utilizan son guiados de la base de datos americana FDA (Food and Drug Administration) una de las bases de datos alimenticias mas extensas que existe, pero no sería preciso guiarse de estos datos ya que los alimentos en Europa por la leyes de agricultura y procesamiento de alimentos los valores alimenticios son distintos, por eso nos guiaremos por los valores de la Base de Datos Española BEDCA (Base de Datos Española de Composición de Alimentos).

Es una red de Centros de investigación públicos, Administración e Instituciones privadas cuyo objetivo es el desarrollo y mantenimiento de la Base de Datos Española de Composición de Alimentos.

Esta Base de datos está construida con los estándares europeos desarrollados por la Red de Excelencia Europea EuroFIR y se incorporará a otras Bases de Datos Europeas dentro la Asociación EuroFIR AISBL encargadas de elaborar una plataforma unificada y con estándares de calidad de las Bases de Datos de Composición de Alimentos Europeas y su interconexión a través de servicios WEB.

La RedBEDCA se ha constituido con una Ayuda para Acciones Complementarias del Ministerio de Ciencia e Innovación (MICINN) y con la financiación y coordinación de la AESAN del Ministerio de Sanidad, Servicios Sociales e Igualdad.

1.2 OBJETIVOS DEL PROYECTO

El objetivo principal es desarrollar una aplicación Web que sirva de apoyo de apoyo para relación entre nutricionista, paciente y comidas, que sea económico y sencillo de utilizar para pacientes y doctor. Para conseguir este objetivo principal hemos seguido una serie de sub-objetivos:

- Una toma de requisitos con un Nutricionista así realmente se captarán los requisitos necesarios para desarrollar la aplicación.
- Decisión en que plataforma se va a desarrollar la aplicación web.
- Decisión en que plataforma de hosting se va a publicar la aplicación.
- Diseño del sistema: se ha realizado un diseño sobre cómo va a ser el sistema viendo que tareas realiza la aplicación.
- Implementación de la aplicación
- Realización de pruebas: para comprobar el funcionamiento se han realizado varias pruebas de la detección de las irregularidades.

1.3 ESTRUCTURA DE LA MEMORIA.

El contenido que se presenta en la memoria de este proyecto de fin de grado está formado por cuatro bloques diferenciados.

En el primer bloque nos encontramos con la introducción al proyecto, una descripción sobre el planteamiento y estructura del proyecto y una justificación de la realización del proyecto y los objetivos a seguir por el mismo.

En el segundo bloque nos encontramos con el marco teórico del proyecto. En esta parte se exponen los temas principales sobre los que trabaja el proyecto. Este bloque está a su vez subdividido en tres partes coincidentes con los tres temas tratados. Los temas que forman este marco práctico son: una comparativa de desarrollo Web y desarrollo Móvil, las razones de porque se ha elegido Django web framework, comparativa entre Amazon Web Services y otros servicios de hosting, las aplicaciones Web, y la importancia de llevar un control alimenticio.

En el tercer bloque se presenta el marco práctico el cual realiza una estructuración más a fondo del proyecto, así como de la aplicación Web que conforma dicho proyecto. En él se describen los requisitos de la aplicación y metodología seguida. También se define el diagrama de entidad relación para la aplicación, la explicación de cada entidad, muestra de la aplicación en producción.

Por último, el cuarto bloque lo forma el manual de usuario que servirá de guía al usuario sobre cómo usar la aplicación

MARCO TEÓRICO

MARCO TEÓRICO

2.1 TEMA 1: DESARROLLO WEB VS MÓVIL

Aunque el desarrollo móvil hoy en día está un auge inmenso es necesario hacer un análisis de que nos conviene para el desarrollo de esta aplicación. Desarrollar una aplicación para móviles implica una planificación elaborada y varios procesos que se unen para formar un todo armónico. Todo comienza con una idea de aplicación, luego pasa a la planificación, el diseño de aplicaciones, el desarrollo de aplicaciones, las pruebas y, finalmente, el despliegue de la aplicación en el dispositivo o dispositivos móviles previstos. Sin embargo, hay una cosa que usted necesita para decidir incluso antes de pasar por las etapas antes mencionadas de desarrollo de aplicaciones.

Antes de decidir sobre que opción utilizar. Hare una comparativa entre aplicación web y aplicación nativa o móvil.

¿Qué son las aplicaciones nativas y Web y cómo son diferentes entre sí? ¿Qué alternativa sería mejor para usted? Aquí hay una comparación entre aplicaciones nativas y aplicaciones web.

Aplicaciones nativas vs. aplicaciones móviles

Una aplicación nativa es una aplicación desarrollada esencialmente para un dispositivo móvil en particular y se instala directamente en el propio dispositivo. Los usuarios de aplicaciones nativas generalmente las descargan a través de las tiendas de aplicaciones en línea o del mercado de aplicaciones, como la App Store de Apple, Google Play Store, etc. Un ejemplo de aplicación nativa es la aplicación Camera + para los dispositivos iOS de Apple.

Una aplicación web, por otro lado, son básicamente aplicaciones habilitadas para Internet accesibles a través del navegador web del dispositivo móvil.

No es necesario descargarlos en el dispositivo móvil del usuario para poder acceder a ellos. El navegador de Safari es un buen ejemplo de una aplicación web para móviles.

Desarrollo de aplicaciones para móviles: Aspectos a tener en cuenta

Aquí hay una comparación rápida entre aplicaciones nativas y aplicaciones web:

Interfaz de usuario

Desde el punto de vista del usuario del dispositivo móvil, algunas aplicaciones nativas y Web se ven y funcionan de la misma manera, con muy poca diferencia entre ellas. La elección entre estos dos tipos de aplicaciones debe hacerse sólo cuando se tiene que decidir si se debe desarrollar una aplicación centrada en el usuario o una aplicación centrada en la aplicación. Algunas empresas desarrollan aplicaciones nativas y Web, con el fin de ampliar el alcance de sus aplicaciones, mientras que también proporcionan una buena experiencia general del usuario.

Proceso de desarrollo de aplicaciones

El proceso de desarrollo de aplicaciones de estos dos tipos de aplicaciones es lo que los distingue entre sí.

Cada plataforma móvil para la que se desarrolla la aplicación nativa, estipula su propio proceso de desarrollo único. En el caso de las aplicaciones Web que se ejecutan en el navegador Web de un dispositivo móvil, el problema que surge es que cada uno de estos dispositivos móviles tienen características únicas y vienen con sus problemas únicos también.

Cada plataforma móvil utiliza un lenguaje de programación nativo diferente. Mientras que iOS utiliza Objective-C, Android utiliza Java, Windows Mobile utiliza C ++ y así sucesivamente. Por otro lado, las aplicaciones web utilizan idiomas como JavaScript, HTML 5, CSS3 u otros marcos de aplicaciones Web según las preferencias del desarrollador.

Cada plataforma móvil ofrece al desarrollador su propio SDK estandarizado, herramientas de desarrollo y otros elementos de interfaz de usuario, que pueden utilizar para desarrollar su aplicación nativa con relativa facilidad. En el caso de las aplicaciones Web, sin embargo, no existe tal normalización y el desarrollador no tiene acceso a SDKs o herramientas de ningún tipo.

Por supuesto, existen varias herramientas y marcos disponibles para el desarrollador, con los que pueden implementar aplicaciones en múltiples plataformas móviles y navegadores Web.

Herramientas de formato de aplicación de multiplataforma DIY y desarrollo de aplicaciones
Accesibilidad

Una aplicación nativa es totalmente compatible con el hardware del dispositivo y características nativas, como acelerómetro, cámara y así sucesivamente.

Las aplicaciones web, por otro lado, sólo pueden acceder a una cantidad limitada de características nativas de un dispositivo.

Mientras una aplicación nativa funciona como una entidad independiente, el problema es que el usuario tiene que seguir descargando actualizaciones. Por el contrario, una aplicación Web se actualiza sin necesidad de intervención del usuario. Sin embargo, necesariamente se debe tener acceso a través del navegador de un dispositivo móvil.

Ganar dinero en aplicaciones

La monetización de aplicaciones con aplicaciones nativas puede resultar complicada, ya que algunos fabricantes de dispositivos móviles pueden imponer restricciones a la integración de servicios con determinadas plataformas y redes de anuncios para móviles. Por el contrario, las aplicaciones web le permiten monetizar aplicaciones a través de anuncios, cobrar cuotas de membresía y así sucesivamente. Sin embargo, mientras la tienda de aplicaciones se encarga de sus ingresos y comisiones en el caso de la aplicación nativa, debe configurar su propio sistema de pago en caso de una aplicación Web.

Eficiencia

Las aplicaciones nativas son más costosas de desarrollarse. Sin embargo, son más rápidos y más eficientes, ya que trabajan en tándem con el dispositivo móvil para el que son desarrollados. Además, están seguros de calidad, ya que los usuarios pueden acceder a ellos sólo a través de las tiendas de aplicaciones en línea.

Las aplicaciones Web pueden provocar mayores costos de mantenimiento en varias plataformas móviles. Además, no existe una autoridad reguladora específica para controlar los estándares de calidad de estas aplicaciones. Sin embargo, la App Store de Apple incluye un listado de las aplicaciones Web de Apple.

Conclusión

Luego de haber tomado en cuenta las necesidades de esta aplicación he decidido que el desarrollo de una aplicación web es la decisión correcto por presupuesto ya que el desarrollo es móvil conlleva un mayor gasto y el mercado de esta aplicación no es masivo el uso de una aplicación web será suficiente para ello.

2.1.1 DJANGO WEB FRAMEWORK.

En los primeros días, los desarrolladores web escribían cada página a mano. Actualizar un sitio Web significaba editar HTML; un "rediseño" implicaba rehacer cada página, una a la vez. A medida que los sitios Web crecieron y se volvieron más ambiciosos, rápidamente se hizo evidente que ese enfoque era tedioso, lento y, finalmente, insostenible.

Un grupo de hackers emprendedores en NCSA (el Centro Nacional para Aplicaciones de Supercomputación, donde se desarrolló Mosaic, el primer navegador web gráfico) resolvió este problema permitiendo que el servidor web generara programas externos que pudieran generar dinámicamente HTML. Llamaron a este protocolo Common Gateway Interface, o CGI, y cambió la Web para siempre. Es difícil ahora imaginar lo que una revelación CGI debe haber sido: en lugar de tratar las páginas HTML como archivos simples en el disco, CGI le permite pensar en sus páginas como recursos generados dinámicamente a demanda.

El desarrollo de CGI marcó el comienzo de la primera generación de sitios Web dinámicos. Sin embargo, CGI tiene sus problemas: las secuencias de comandos CGI necesitan contener un montón de código repetitivo "cliché", hacen difícil la reutilización de código y pueden ser difíciles para los desarrolladores principiantes escribir y entender.

PHP arregló muchos de estos problemas, y tomó el mundo por asalto - ahora es la herramienta más popular utilizada para crear sitios web dinámicos, y decenas de lenguajes similares (ASP, JSP, etc.) siguieron el diseño de PHP de cerca. La principal innovación de PHP es su facilidad

de uso: el código PHP está simplemente incrustado en HTML simple; la curva de aprendizaje para alguien que ya conoce HTML es extremadamente superficial.

Pero PHP tiene sus propios problemas; su muy facilidad de uso fomenta el código descuidado, repetitivo, mal concebido. Peor aún, PHP hace poco para proteger a los programadores de las vulnerabilidades de seguridad, por lo que muchos desarrolladores de PHP se encontraron aprendiendo sobre la seguridad sólo una vez que era demasiado tarde.

Estas y otras frustraciones similares llevaron directamente al desarrollo de la actual cosecha de marcos de desarrollo Web de "tercera generación". Con esta nueva explosión de desarrollo web viene otro aumento en la ambición; Se espera que los desarrolladores web hagan más y más todos los días.

Django fue inventado para cumplir con estas nuevas ambiciones.

Django está modelado alrededor de un marco Model-View-Controller (MVC). MVC es un patrón de diseño de software que tiene como objetivo separar una aplicación web en tres partes de interconexión:

- El modelo, que proporciona la interfaz con la base de datos que contiene los datos de la aplicación;
- La vista, que decide qué información presentar al usuario y recoge información del usuario; y
- El controlador, que gestiona la lógica de negocio de la aplicación y actúa como intermediario de información entre el modelo y la vista.

Django utiliza una terminología ligeramente diferente en su implementación de MVC (Figura 3-1). En Django:

- El modelo es funcionalmente el mismo. La asignación objeto-relacional de Django (ORM-más en el ORM más adelante) proporciona la interfaz a la base de datos de la aplicación;

- La plantilla proporciona una lógica de visualización y es la interfaz entre el usuario y su aplicación Django; y
- La vista gestiona la mayor parte del procesamiento de datos de las aplicaciones, la lógica de la aplicación y la mensajería.

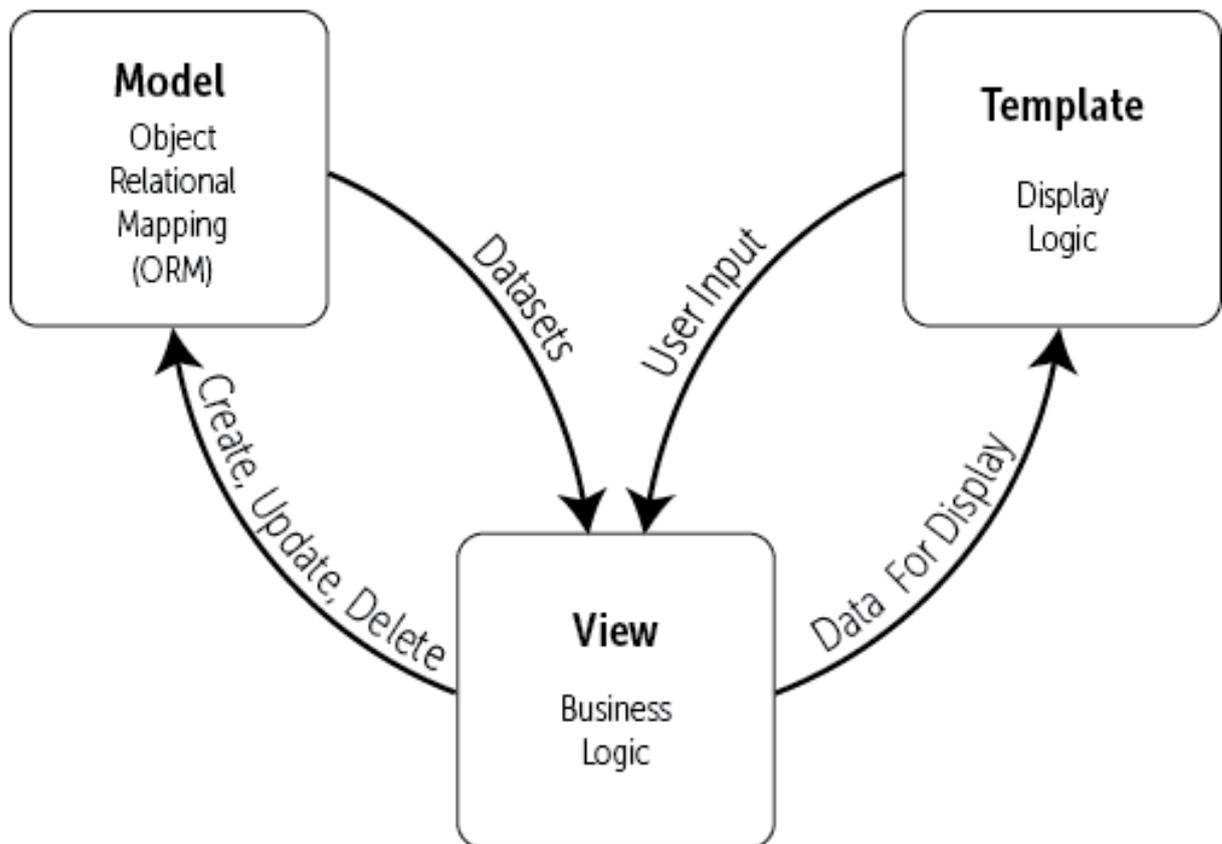


Figura 3-1. Una descripción pictórica del patrón Model-Template-View (MTV) en Django.

El patrón de diseño MVC se ha utilizado tanto para aplicaciones de escritorio como para aplicaciones web durante muchos años, por lo que hay un gran número de variaciones en este tema, de las cuales Django no es una excepción. Si desea profundizar un poco más en el patrón de diseño MVC.

Django Models

Los modelos de Django proporcionan una asignación de objetos (ORM) a la base de datos subyacente. ORM es una potente técnica de programación que facilita mucho el trabajo con bases de datos y bases de datos relacionales.

La mayoría de las bases de datos comunes se programan con algún tipo de lenguaje de consulta estructurado (SQL), sin embargo, cada base de datos implementa SQL a su manera. SQL puede ser bastante complejo y difícil de aprender. Por otro lado, una herramienta ORM proporciona una asignación simple entre un objeto (el 'O' en ORM) y la base de datos subyacente, sin que el programador necesite conocer la estructura de la base de datos o requiera SQL complejo para manipular y recuperar datos. 3-2).

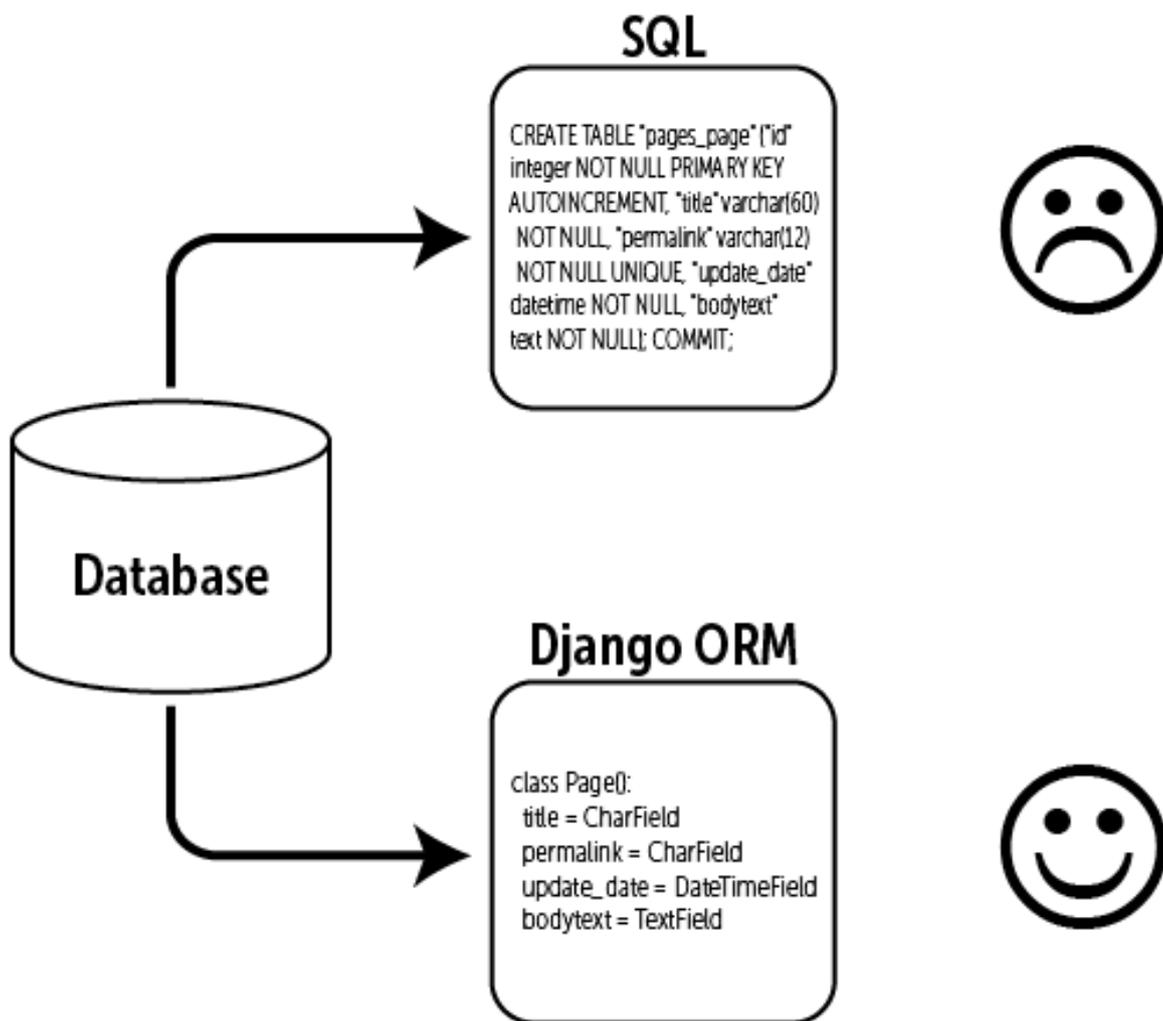


Figura 3-2. Un ORM permite la manipulación simple de los datos sin tener que escribir complejos SQL.

En Django, el modelo es el objeto que se asigna a la base de datos. Cuando crea un modelo, Django crea una tabla correspondiente en la base de datos (Figura 3-3), sin tener que escribir una sola línea de SQL. Django prefijará el nombre de la tabla con el nombre de su aplicación de Django (más información sobre las aplicaciones de Django más adelante).

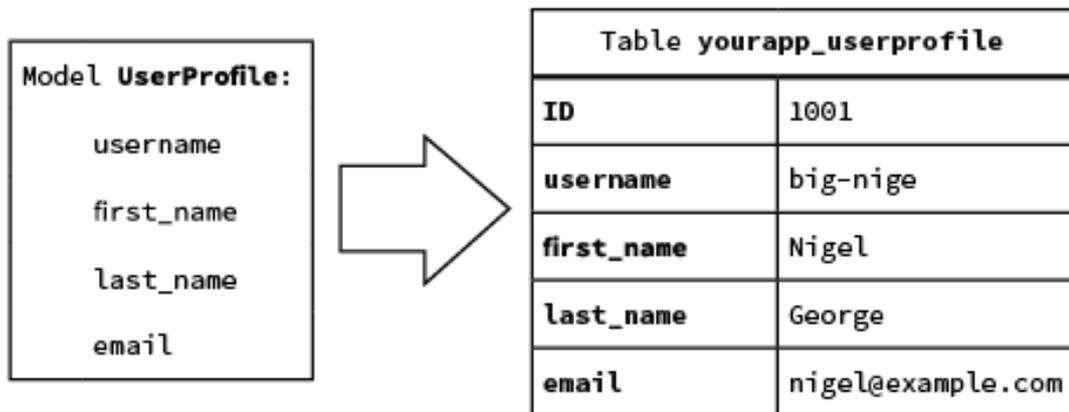


Figura 3-3. La creación de un modelo Django crea una tabla correspondiente en la base de datos.

El modelo también vincula la información relacionada en la base de datos. En la Figura 3-4, se crea un segundo modelo para realizar un seguimiento de los cursos en los que un usuario está inscrito. Repetir la información de todos los usuarios en la tabla `yourapp_Course` estaría en contra de buenos principios de diseño, por lo que en vez de crear una relación en ORM) entre la tabla `yourapp_Course` y la tabla `yourapp_UserProfile`.

Figura 3-4. Las relaciones entre tablas se crean mediante enlaces de clave externa en modelos de Django.

Esta relación se crea mediante la vinculación de los modelos con una clave extranjera, en otras palabras, el campo `user_id` de nuestra tabla `yourapp_Course` es un campo clave que está vinculado al campo `id` de la tabla externa `yourapp_UserProfile`.

Esto es una simplificación, pero es una visión general práctica de cómo el ORM de Django usa los datos del modelo para crear tablas de base de datos.

Bases de datos Soportadas

Django admite oficialmente cuatro bases de datos:

- PostgreSQL
- MySQL
- SQLite
- Oracle

También hay varios paquetes de terceros para conectarse a otras bases de datos, si necesita conectarse a una base de datos no oficialmente soportada.

Django Templates

Una plantilla de Django es un archivo de texto diseñado para separar los datos de una aplicación de la forma en que se presenta. En la mayoría de los casos, las plantillas de Django son archivos HTML (Hypertext Markup Language) para presentar los datos de la aplicación en un navegador web, sin embargo las plantillas de Django no se limitan a HTML. Pueden usarse para renderizar varios formatos de texto.

El diseño de las plantillas de Django se basa en varios principios fundamentales, sin embargo, tres son claves:

- Un sistema de plantilla debe separar la lógica del programa del diseño.
- Las plantillas deben desincentivar la redundancia-No repetirse (DRY).
- El sistema de plantilla debe ser seguro y la ejecución de código segura en la plantilla debe estar prohibida.

Lógica separada del diseño

Diseño web y programación web son dos disciplinas muy diferentes. Para todos menos los proyectos más pequeños, el diseño y la programación no son realizados por las mismas personas; en muchos casos, ni siquiera la misma empresa.

Cuando los creadores de Django consideraron primero el diseño del sistema de plantillas de Django, quedó claro que los programadores de Django y los diseñadores de sitios web deben ser capaces de trabajar independientemente entre sí. El resultado es un lenguaje de secuencias de comandos de texto sin formato que utiliza etiquetas para proporcionar una lógica de presentación para decidir qué contenido mostrar en la plantilla. Esto es más fácil de entender con un simple ejemplo:

```
<h1>Your Order Information</h1>  
<p>Dear {{ person_name }},</p>
```

Esta podría ser la primera pareja de líneas de una página de confirmación de pedido, que se muestran en un sitio web después de que el usuario haya realizado una compra. Notará que la mayoría de este código es HTML normal. El pequeño bit de secuencia de comandos en negrita es una etiqueta de variable de Django. Cuando se muestre esta plantilla en su navegador, la plantilla reemplazará la variable `nombre_persona` con el nombre pasado a la plantilla por la vista.

Como se trata de texto sin formato y HTML, un diseñador no necesita saber nada acerca de Django para poder crear una plantilla de Django. Todo lo que el diseñador tiene que hacer es añadir un marcador de posición (etiqueta de comentario HTML, por ejemplo), para que el programador reemplace con una etiqueta Django al codificar el sitio web.

La otra gran ventaja de este enfoque es que, dado que la mayor parte de la plantilla es HTML simple, como programador, puede crear un sitio web atractivo sin un diseñador, puede descargar una plantilla HTML de Internet y agregar etiquetas de plantilla de Django. Esto también funciona con las plantillas Bootstrap y los sitios pesados en el front-end de JavaScript.

Don't Repeat Yourself (DRY)

DRY es un término que aparece con frecuencia en las discusiones de Django, ya que es uno de los principios centrales de Django. El principio DRY es particularmente evidente en cómo Django usa la herencia de la plantilla. Para entender mejor cómo la herencia de plantilla nos ayuda a minimizar la repetición y el código redundante, examinemos primero un diseño de página web típico (Figura 3-5).

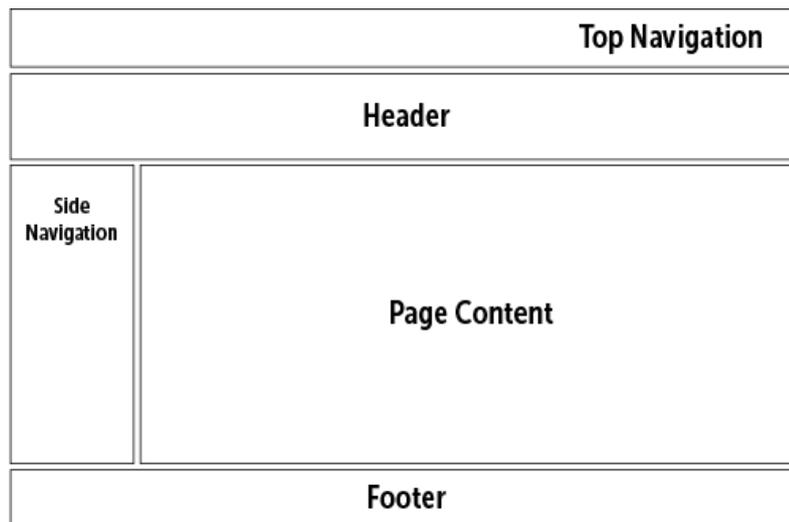
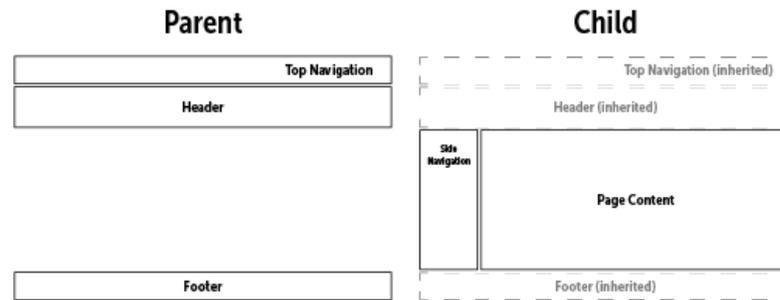


Figura 3-5. Un diseño de páginas web típico con elementos comunes como encabezado, pie de página y navegación.

Este diseño de página tiene una navegación superior, una imagen de encabezado, menú del lado izquierdo, el contenido principal de la página y un pie de página. Si sólo quería crear algunas páginas web, podría evitar copiar la página principal y simplemente cambiar el contenido y guardar cada página diferente como un archivo HTML.

El problema es que, no sólo estamos repitiendo un montón de código, sino que mantener un sitio grande podría salir rápidamente de las manos.

Django soluciona este problema creando una plantilla padre que tiene el contenido que es común a todo el sitio web y luego creamos plantillas secundarias que heredan estas características comunes y luego agregamos cualquier contenido exclusivo para la plantilla secundaria (Figura 3-6).



Se puede ver como se incluye la navegación de la barra lateral en el niño aquí. Es común que ciertas páginas de un sitio tengan navegación limitada, por lo que no todas las páginas necesitarán la navegación lateral.

Django admite múltiples herencias también, siguiendo el ejemplo anterior, podría tener una plantilla secundaria que añada sólo la navegación lateral al padre y luego tener una tercera plantilla que hereda del niño y añada el contenido.

El único límite en cómo cortar y cortar la herencia de plantilla de Django es práctico: si tienes plantillas que heredan más de dos o tres de profundidad, debes reevaluar el diseño de tu sitio.

Seguridad de las Plantillas (templates)

La filosofía de Django es que Internet es lo suficientemente insegura, sin introducir problemas de seguridad al permitir la ejecución de código en plantillas de páginas web. La solución de Django para plantillas de vulnerabilidades de seguridad es simple: la ejecución de código está prohibida.

Las etiquetas de plantilla de Django proporcionan sólo lógica de visualización, esto incluye:

- Visualización de variables: puede ser texto simple como un nombre de usuario o datos más complejos como texto con formato HTML.
- Elección del contenido que se mostrará en función de comprobaciones lógicas, por ejemplo, si un usuario está conectado, luego mostrará el menú de usuario o el contenido de sólo usuario.
- Iterando sobre listas de datos, lo más a menudo utilizado para insertar información de base de datos en listas HTML.

- Formato de los datos: para el formato de la fecha, la manipulación de texto y otros filtros que actúan sobre los datos.

Cosas que no puedes hacer en una plantilla de Django:

- Ejecutar código Python
- Asignar un valor a una variable
- Realizar una lógica avanzada

Las plantillas de Django también añaden funciones de seguridad adicionales como el escape automático de todas las cadenas, Cross-Site Scripting y Cross-Site Request Forgery. Estos dos últimos temas están más allá de un texto inicial, pero es útil entender que las plantillas de Django son seguras por defecto, por lo que no tiene que preocuparse por introducir problemas de seguridad en su sitio web accidentalmente.

Vistas de Django

Los puntos de vista de Django son los intermediarios de información de una aplicación de Django. Una vista genera datos de su base de datos (o una fuente de datos externa o servicio) y la entrega a una plantilla. Para una aplicación web, la vista ofrece contenido de página web y plantillas, para una API RESTful, este contenido podría tener datos JSON correctamente formateados.

La vista toma decisiones sobre qué datos se entregan a la plantilla, ya sea actuando sobre la entrada del usuario o en respuesta a otra lógica de negocio y procesos internos.

Cada vista de Django realiza una función específica y tiene una plantilla asociada. Las vistas están representadas por una función Python o un método de una clase Python. En los primeros días de Django, sólo había vistas basadas en funciones, sin embargo, como Django ha crecido a lo largo de los años, los desarrolladores de Django agregaron vistas basadas en la clase a Django.

Las vistas basadas en clases añaden extensibilidad a las vistas de Django, así como vistas integradas que hacen que la creación de vistas comunes (como mostrar una lista de artículos)

sea más fácil de implementar. No se preocupe demasiado por las diferencias entre las vistas basadas en funciones y en las clases ahora, cubriremos ambas en más detalle más adelante en el libro.

Para aliviar la carga de los programadores, muchas tareas comunes de visualización tienen vistas integradas en Django. Existen cuatro vistas basadas en funciones para mostrar páginas de error:

- La página 404 (página no encontrada)
- La vista 500 (error de servidor)
- La vista 403 (HTTP Prohibida)
- The 400 (bad request) ver más

También hay varias vistas basadas en clases para simplificar tareas comunes de visualización. Incluyen:

- ListView para mostrar una lista de objetos de datos (por ejemplo, enumerar todos los artículos)
- DetailView para mostrar un solo objeto (por ejemplo, un artículo individual)
- RedirectView redirecciona a otra URL
- FormView para mostrar un formulario

También se ofrecen vistas adicionales de las fechas genéricas basadas en clases para mostrar colecciones de objetos, como publicaciones del día, semana y mes y anuales.

URLconf

Nuestro sitio web no es de mucha utilidad si no podemos navegar a nuestro alrededor-tenemos que decirle a la vista qué mostrar en el navegador, en función de lo que el usuario ha solicitado.

La navegación en un sitio web de Django es la misma que cualquier otra página web y se accede a través de URL. Cuando un usuario hace clic en un enlace de un sitio web, se envía una solicitud para esa URL a Django (Figura 3-7).

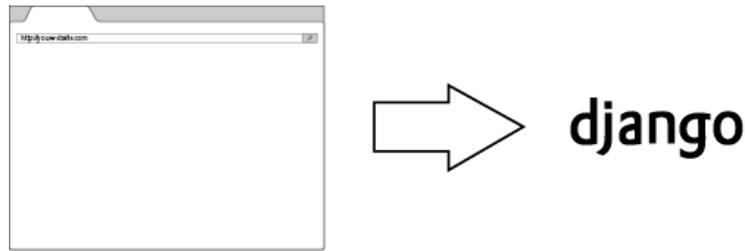


Figura 3-7. La solicitud del navegador para su sitio web se envía directamente a Django.

Una vez que Django recibe la URL solicitada, debe decidir qué vista tratará la solicitud. Usted, como programador, decide qué vista servir en qué URL creando una Configuración de URL (URLconf para abreviar) en un archivo de Python denominado `urls.py` (Figura 3-8). Cuando Django encuentra una URL en `urls.py` que coincide con la URL solicitada, llama a la vista asociada con esa URL.

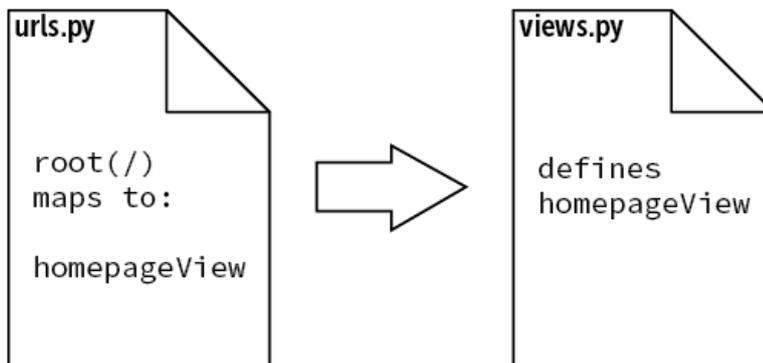


Figura 3-8. Django asigna la URL solicitada a una vista.

La vista seleccionada vuelve a convertir el contenido en una plantilla, según la lógica de negocio de la vista, y envía el contenido renderizado de nuevo a su navegador para su visualización (Figura 3-9).

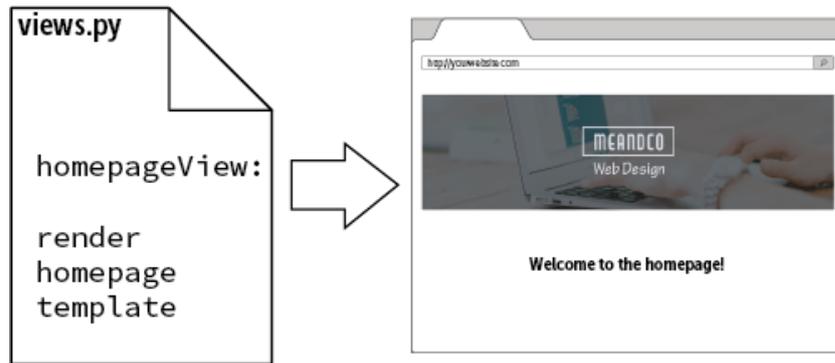


Figura 3-9. A continuación, la vista le indica a Django qué plantilla utilizar al enviar el contenido al explorador.

Por supuesto, esto es una simplificación: Django puede recopilar datos mucho más complejos del navegador que una URL, y las vistas no sólo representan páginas web. También hay otra capa de Django que se encuentra entre la solicitud del navegador y la respuesta de su vista (que Django llama bastante astutamente middleware) que le permite hacer toneladas de cosas interesantes con sus datos.

2.1.2 ELECCIÓN DE DJANGO

Aunque la forma en que Django logra muchas tareas complejas de una manera tan simple podría considerarse muy inteligente, el enfoque principal de Django en hacer que las cosas se hagan se cuece en su ADN.

Django tiene sus debilidades, pero su acercamiento pragmático a conseguir el producto final completado es donde realmente se destaca de los demás framework por algo su eslogan es “The web framework for perfectionists with deadlines” (el framework web para perfeccionista con fechas de entrega” . Django tiene un montón de partidarios y algunos detractores, sin embargo, para lo que a mi respecta este es el Top 10 de Django:

1. Python
2. Baterías incluidas
3. No se interpone en tu camino
4. Administrador incorporado
5. Escalable
6. Probado en batalla
7. Variedad de paquetes
8. Desarrollado activamente
9. Lanzamientos estables
10. Documentación de primera clase

1. Python

Python es sin duda el lenguaje de programación más fácil de aprender.

Con su uso de construcciones de lenguaje natural (por ejemplo, diseño de párrafos y sangría) y sintaxis sencilla de aprender, Python hace que entender la estructura del programa y el flujo mucho más fácil de aprender que otros lenguajes populares.

Esto es evidente en el hecho de que la mayor proporción de cursos de programación introductoria en las universidades y los colegios ahora utilizan Python como el idioma de elección.

Python hoy tiene múltiples implementaciones incluyendo Jython, scripted en lenguaje Java para Java Virtual Machine; IronPython escrito en C # para la Common Language Infrastructure, y la versión PyPy escrita en RPython y traducida a C. A destacar, Cpython que está escrito en C y desarrollado por Python Software Foundation es la implementación por defecto y más popular de Python. Si bien estas implementaciones funcionan en el idioma nativo en el que están escritas, también son capaces de interactuar con otros idiomas a través del uso de módulos. La mayoría de estos módulos trabajan en el modelo de desarrollo comunitario y son de código abierto y gratuitos.

Ventajas / Beneficios de Python

La aplicación diversa del lenguaje Python es el resultado de la combinación de características que dan a este lenguaje una ventaja sobre otros. Algunos de los beneficios de la programación en Python incluyen:

- Presencia de Módulos de Terceros:

El Python Package Index (PyPI) contiene numerosos módulos de terceros que hacen que Python sea capaz de interactuar con la mayoría de los otros idiomas y plataformas.

- Extensas bibliotecas de apoyo:

Python proporciona una gran biblioteca estándar que incluye áreas como protocolos de Internet, operaciones de cadena, herramientas de servicios web y interfaces de sistemas operativos. Muchas tareas de programación de alto uso ya han sido escritas en la biblioteca estándar, lo que reduce la longitud del código que se escribirá de manera significativa.

- Desarrollo de código abierto y comunidad:

El lenguaje Python se desarrolla bajo una licencia de código abierto aprobada por OSI, lo que la hace gratuita para usar y distribuir, incluso para propósitos comerciales.

Además, su desarrollo es impulsado por la comunidad que colabora para su código a través de conferencias de hosting y listas de correo, y proporciona sus numerosos módulos.

- Facilidad de aprendizaje y soporte disponibles:

Python ofrece una excelente facilidad de lectura y una sencilla sintaxis sencilla de aprender que ayuda a los principiantes a utilizar este lenguaje de programación. Las directrices de estilo de código, PEP 8, proporcionan un conjunto de reglas para facilitar el formato de código. Además, la amplia base de usuarios y desarrolladores activos ha dado lugar a un rico banco de recursos de Internet para fomentar el desarrollo y la adopción continua del lenguaje.

- Estructuras de datos fáciles de usar:

Python incorpora estructuras de datos de listas y diccionarios que pueden usarse para construir estructuras de datos de tiempo de ejecución rápidas. Además, Python también ofrece la opción de escribir datos dinámicos de alto nivel, lo que reduce la longitud del código de soporte que se necesita.

- Productividad y velocidad:

Python tiene un diseño limpio orientado a objetos, proporciona capacidades mejoradas de control de procesos y posee fuertes capacidades de integración y procesamiento de texto y su propio marco de pruebas unitarias, todo lo cual contribuye al aumento de su velocidad y productividad. Python se considera una opción viable para la construcción de complejas aplicaciones de red multiprotocolo.

Aplicaciones de Python

- Aplicaciones de escritorio basadas en GUI:

Python tiene una sintaxis sencilla, arquitectura modular, herramientas de procesamiento de texto enriquecido y la capacidad de trabajar en múltiples sistemas operativos que lo convierten en una opción deseable para el desarrollo de aplicaciones basadas en escritorio. Hay varios kits de herramientas GUI como wxPython, PyQt o PyGtk disponibles que ayudan a los desarrolladores a crear una interfaz gráfica de usuario altamente funcional (GUI). Las diversas aplicaciones desarrolladas con Python incluyen:

Procesamiento de imágenes y aplicaciones de diseño gráfico:

Python se ha utilizado para crear software de imagen 2D como Inkscape, GIMP, Paint Shop Pro y Scribus. Además, los paquetes de animación 3D, como Blender, 3ds Max, Cinema 4D, Houdini, Lightwave y Maya, también utilizan Python en proporciones variables.

Aplicaciones Científicas y Computacionales:

Las velocidades más altas, la productividad y la disponibilidad de herramientas, como Scientific Python y Numeric Python, han dado como resultado que Python se convierta en una parte integral de las aplicaciones involucradas en la computación y procesamiento de datos científicos. El software de modelado 3D, como FreeCAD, y el software de métodos de elementos finitos, como Abaqus, están codificados en Python.

Juegos:

Python tiene varios módulos, bibliotecas y plataformas que soportan el desarrollo de juegos. Por ejemplo, PySoy es un motor de juegos 3D que soporta Python 3, y PyGame proporciona funcionalidad y una biblioteca para el desarrollo de juegos. Ha habido numerosos juegos construidos utilizando Python incluyendo Civilization-IV, Disney Toontown Online, Vega Strike, etc.

- Framework Web y Aplicaciones Web:

Python ha sido utilizado para crear una variedad de web-frameworks incluyendo CherryPy, Django, TurboGears, Bottle, Flask, etc. Estos marcos proporcionan bibliotecas y módulos estándar que simplifican las tareas relacionadas con la gestión de contenidos, la interacción

con la base de datos y la interfaz con diferentes protocolos de Internet como HTTP, SMTP, XML-RPC, FTP y POP. Plone, un sistema de gestión de contenidos; ERP5, un ERP de código abierto que se utiliza en el sector aeroespacial, la confección y la banca; Odoo - un conjunto consolidado de aplicaciones empresariales; y el motor de Google App son algunas de las aplicaciones web más populares basadas en Python.

- Aplicaciones Empresariales y Empresariales:

Con características que incluyen bibliotecas especiales, extensibilidad, escalabilidad y sintaxis fácilmente legible, Python es un lenguaje de codificación adecuado para personalizar aplicaciones más grandes. Reddit, que fue escrito originalmente en Common Lisp, fue reescrito en Python en 2005. Python también contribuyó en gran parte a la funcionalidad en YouTube.

- Sistemas operativos:

Python es a menudo una parte integral de las distribuciones de Linux. Por ejemplo, el Ubiquity Installer de Ubuntu y el Anaconda Installer de Fedora y Red Hat Enterprise Linux están escritos en Python. Gentoo Linux hace uso de Python para Portage, su sistema de gestión de paquetes.

- Desarrollo del lenguaje:

La arquitectura de diseño y módulo de Python ha influido en el desarrollo de numerosos idiomas. El lenguaje Boo utiliza un modelo de objeto, sintaxis y sangría, similar a Python. Además, la sintaxis de lenguajes como Swift, CoffeeScript, Cobra y OCaml de Apple comparten similitud con Python.

- Prototipado:

Además de ser rápido y fácil de aprender, Python también tiene la ventaja de código abierto de ser libre con el apoyo de una comunidad grande. Esto la convierte en la opción preferida para el desarrollo de prototipos. Además, la agilidad, la extensibilidad y la escalabilidad y la

facilidad del código de refactorización asociado con Python permiten un desarrollo más rápido desde el prototipo inicial.

Aprender Python casi seguramente le servirá no importa donde su carrera le lleva. Python es popular. Google, una de las mayores empresas del mundo, utiliza Python en muchas de sus aplicaciones. También es ampliamente utilizado por programadores profesionales.

2. Baterías incluidas

Django hereda su filosofía de "baterías incluidas" de Python.

Esto a menudo se interpreta como el significado de que Django incluye una gran cantidad de cosas extra que probablemente no necesitará, sin embargo, la mejor analogía es que, en lugar de tener que abrir el idioma para insertar su propia energía (baterías), sólo tiene que haga clic en el interruptor y Django hace el resto.

En términos prácticos esto significa que Django implementa algunos procesos comunes, pero complejos, proporcionando herramientas simples y envolturas para ocultar la complejidad sin comprometer la potencia.

Las "baterías" de Django se encuentran en los paquetes contrib. Los paquetes contrib son:

- admin— La aplicación de administración de Django
- auth— framework de autenticación de Django
- contenttypes— un framework para enganchar en los modelos de Django
- flatpages— un marco para gestionar páginas de casos especiales, como políticas de sitio y términos y condiciones de uso
- gis— añade capacidades geoespaciales a Django
- humanize— añade filtros de plantillas para mejorar la legibilidad de los datos
- messages— a framework for managing session- and cookie-based messages
- postgres— características específicas de la base de datos postgres-
- redirects— gestiona redirecciones
- sessions— le permite operar varios sitios web desde la instalación de uno
- sites— allows you to operate multiple websites from the one installation
- sitemaps— implementa archivos XML

- syndication—a framework para generar syndication feeds

Los paquetes contrib pueden ser un poco complejos, por lo que sólo tocaremos uno o dos de ellos en este libro, sin embargo, como puede ver, Django proporciona una sólida lista de potentes módulos incorporados para que no tenga que crear usted mismo.

3. No se interpone en tu camino

Cuando se crea una aplicación de Django, Django no agrega ninguna función clásica, cruft o funciones innecesarias. No hay importaciones obligatorias, no hay bibliotecas de terceros requeridas ni archivos de configuración XML.

Esto puede ser un poco aterrador cuando se crea un proyecto de Django, ya que las herramientas automáticas de Django (`startproject` y `startapp`) sólo crean un archivo de configuración básica, unas pocas carpetas y algunos archivos de inicio casi vacíos.

Si bien esto puede parecer una mala cosa, en realidad es un gran beneficio como Django le ha proporcionado una base sólida que puede construir sobre de cualquier manera.

4. Panel de Administración incorporado

Fuera de la caja, Django le proporciona una interfaz de administración para trabajar con sus modelos y administrar usuarios, permisos de usuario y grupos.

La interfaz del modelo sustituye inmediatamente la necesidad de un programa de administración de base de datos independiente para todas las funciones de la base de datos, excepto las avanzadas.

Con cambios muy sencillos en su configuración de administración, se puede organizar los campos de su modelo, mostrar y ocultar campos, ordenar, filtrar y organizar sus datos para maximizar su eficiencia.

El administrador también tiene una característica de documentación de modelo opcional que proporciona documentación automática de sus modelos.

La gestión de usuarios siempre es importante en un sitio web moderno y Django ofrece todo lo que se puede esperar: agregar y modificar usuarios, cambiar contraseñas, crear grupos de usuarios, asignar permisos y comunicarse con los usuarios.

Al igual que el resto de Django, el administrador también es personalizable y extensible.

Por ejemplo, se pueden sustituir las plantillas de visualización de administración y se agrega nueva funcionalidad para tareas como exportar los datos del modelo a un archivo delimitado por comas (CSV).

5. Escalable

Django se basa en el patrón de diseño Model-View-Controller (MVC). Esto significa que la base de datos, código de programa (back-end) y código de visualización (front-end) están separados.

Django lleva esta separación un paso más lejos al separar el código de los medios estáticos - imágenes, archivos, CSS y JavaScript- que componen su sitio.

Estas filosofías de diseño le permiten:

- Ejecute servidores independientes para su base de datos, aplicaciones y medios;
- Facilite la distribución de sus medios desde una red de distribución de contenido (CDN);
- Contenido de caché en múltiples niveles y ámbitos; y
- Para sitios realmente grandes, emplee agrupación y equilibrio de carga para distribuir su sitio web a través de varios servidores.
- Django soporta una amplia gama de vendedores populares de terceros para servidores web, gestión de rendimiento, almacenamiento en caché, agrupación y equilibrio.

También soporta aplicaciones y servicios de correo electrónico y mensajería como OAuth y REST.

6. Probado en batalla

Una buena manera de saber si un marco web es robusto y confiable es saber cuánto tiempo ha estado alrededor, si está creciendo y qué sitios de alto perfil lo están utilizando.

Django fue primero de código abierto en 2005, después de correr durante varios años en el entorno de alta demanda de una organización de noticias.

Después de casi 12 años de crecimiento, Django ahora no solo publica compañías de noticias como el Washington Post, sino que también está ejecutando todo o parte de las principales empresas globales como Pinterest, Instagram, Disqus, Bitbucket, EventBrite y Zapier.

Django sigue creciendo en popularidad. Djangosites lista más de 5200 sitios usando Django, y eso es solo para sitios que se registran con Djangosites.

Sería imposible adivinar cuántas páginas Django sirve todos los días en comparación con otras tecnologías en Internet, pero eso es en gran medida irrelevante-Django ha demostrado su eficacia en los últimos 12 años al ejecutar algunos de los sitios con mayor tráfico en Internet y sigue creciendo su base de usuarios hoy en día.

7. Variedad de Paquetes

Casi cualquier cosa que es probable que desee hacer con Django se ha hecho antes. Muchos de la gran comunidad internacional de desarrolladores de Django devuelven a la comunidad liberando sus proyectos como paquetes de código abierto.

El repositorio más grande de estos proyectos se puede encontrar en el sitio de Django Packages. En el momento de escribir este artículo, Django Packages incluye más de 3400 aplicaciones, sitios y herramientas Django reutilizables para usar en sus propios proyectos Django. Un recorrido rápido de paquetes populares incluye:

- Wagtail, Mezzanine y django CMS sistemas de gestión de contenidos
- Cookiecutter: configuración rápida y fácil de las estructuras de proyectos y aplicaciones de Django para aplicaciones más avanzadas
- Django ReST Framework: implementa una API ReST en Django
- Autenticación de Django allauth-Facebook, GitHub, Google y Twitter para tus aplicaciones Django
- Barra de herramientas de depuración: muestra información de depuración mientras se ejecuta el proyecto
- Django Celery-ofrece integración de celery para Django
- Oscar, Django Shop y Cartridge-eCommerce marcos para Django (Cartucho es una extensión para Mezzanine CMS)

Con miles de paquetes más como estos, es muy probable que encuentre un paquete que se adapte a sus necesidades, sin tener que reinventar la rueda.

8. Desarrollado activamente

Uno de los mayores riesgos de código abierto es si hay suficiente interés en el proyecto para atraer el apoyo de desarrolladores en el largo plazo.

No existe tal riesgo con Django, no sólo el proyecto tiene más de 12 años, tiene una larga historia de lanzamientos consistentes y continúa siendo apoyado por una comunidad activa y un gran equipo central de colaboradores voluntarios que mantienen y mejoran el código base cada día.

Django tuvo su primer lanzamiento de producción en 2008 (versión 1.0) y ha tenido tres LTS (Long Term Support), versiones 1.4, 1.8 y 1.11. La versión 1.11 LTS (versión actual) tiene un soporte completo hasta mediados de 2020.

El equipo de desarrollo de Django mantiene una hoja de ruta de desarrollo en el sitio web del proyecto Django y tiene un sólido historial de cumplimiento de los hitos de la hoja de ruta.

El proyecto Django también cuenta con el apoyo de una fundación independiente -la Fundación de Software Django- que es una organización sin fines de lucro registrada en los Estados Unidos.

9. Lanzamientos estables

Los proyectos de software de código abierto son, en muchos casos, más activamente desarrollados y más seguros que el software propietario de la competencia.

La desventaja del desarrollo en constante evolución de un proyecto de software de código abierto es la falta de una base de código estable en la que basar el desarrollo comercial.

Django aborda este problema con versiones de soporte a largo plazo (LTS) del software y un proceso de liberación definido.

Las versiones LTS se liberan con un período de soporte garantizado (típicamente tres años). En este período se garantiza que la base de código se mantenga estable; con parches para errores, seguridad y pérdida de datos 100% compatible con la versión de características.

El proceso de lanzamiento de Django garantiza que las versiones oficiales sean lo más estables posible. Después de una fase de desarrollo, cada liberación entra en una fase Alpha donde se aplica una característica congelada.

La nueva versión pasa a través de las etapas de Beta y Release Candidate (RC) donde se solucionan los errores del lanzamiento. Si no se encuentran errores importantes durante un período posterior al lanzamiento del candidato, se lanzará el final (liberación de funciones).

Después de que se haya liberado el final, sólo se aplicarán las correcciones de errores y parches de seguridad. Estos parches, al igual que las versiones LTS, son 100% compatibles con el lanzamiento de características.

10. Documentación de primera clase

Incluso en los primeros lanzamientos, los desarrolladores de Django se aseguraron de que la documentación fuera completa y que los tutoriales fueran fáciles de seguir.

Para mí, la documentación debería ser el número uno en esta lista porque era la calidad de la documentación que me hizo elegir Django sobre otras opciones.

Esto fue en 2007, cuando Django todavía era la versión 0.96 antes de que hubiera hecho su primer lanzamiento oficial. La documentación oficial sólo ha mejorado en los años intermedios.

Django también tiene un fuerte apoyo de miembros de la comunidad que producen materiales de aprendizaje gratuitos, libros, cursos gratuitos y pagados y un montón de consejos, trucos y asistencia en sus sitios web.

Por todos estos puntos he elegido Django sobre cualquier otro framework web también tomando en cuenta que es un framework web de tercera generación.

2.2. TEMA 2: AMAZON WEB SERVICES

2.2.1 AMAZON WEB SERVICES.

AWS Cloud ofrece un amplio conjunto de servicios de infraestructura, como potencia de cómputo, opciones de almacenamiento, redes y bases de datos que se entregan como una utilidad: bajo demanda, disponible en segundos, con un sistema de pago por uso. Desde el almacenamiento de datos hasta las herramientas de implementación, los directorios para la entrega de contenido, más de 90 servicios AWS están disponibles. Los nuevos servicios se pueden aprovisionar rápidamente, sin gastos iniciales de capital. Esto permite a las empresas, las empresas de nueva creación, las pequeñas y medianas empresas y los clientes del sector público acceder a los bloques de construcción que necesitan para responder rápidamente a las necesidades cambiantes del negocio. Este documento técnico le ofrece una visión general de los beneficios de AWS Cloud y le presenta los servicios que componen la plataforma.

En 2006, Amazon Web Services (AWS) comenzó a ofrecer servicios de infraestructura de TI a las empresas en forma de servicios web, ahora conocidos comúnmente como cloud computing. Uno de los beneficios clave de la computación en la nube es la oportunidad de reemplazar los gastos iniciales de infraestructura de capital con bajos costos variables que se ajustan a su negocio. Con la nube, las empresas ya no necesitan planificar y adquirir servidores y otras infraestructuras de TI semanas o meses de antelación. En su lugar, pueden girar instantáneamente cientos o miles de servidores en minutos y entregar resultados más rápido.

Hoy en día, AWS proporciona una plataforma de infraestructura de bajo costo, escalable y altamente confiable en la nube, que impulsa a cientos de miles de empresas en 190 países de todo el mundo.

¿Qué es la computación en la nube?

La computación en nube es la entrega bajo demanda de potencia de computación, almacenamiento de bases de datos, aplicaciones y otros recursos de TI a través de una plataforma de servicios en la nube a través de Internet con un sistema de pago por uso. Ya sea que esté ejecutando aplicaciones que compartan fotos con millones de usuarios móviles o esté dando soporte a las operaciones críticas de su empresa, una plataforma de servicios en la nube proporciona acceso rápido a recursos de TI flexibles y de bajo costo. Con la computación en la nube, no es necesario hacer grandes inversiones iniciales en hardware y pasar mucho tiempo en el trabajo pesado de la gestión de ese hardware. En su lugar, puede proporcionar exactamente el tipo y el tamaño adecuado de los recursos informáticos que necesita para alimentar su nueva idea brillante u operar su departamento de TI. Puedes acceder a tantos recursos como necesites, casi al instante, y solo pagas por lo que usas.

Seis ventajas de la computación en la nube

- Gasto de capital comercial para gastos variables: En lugar de tener que invertir mucho en centros de datos y servidores antes de saber cómo va a utilizarlos, sólo puede pagar cuando consume recursos informáticos y pague sólo por lo que consume.
- Beneficio de enormes economías de escala. Mediante el uso de la computación en nube, podemos lograr un costo variable más bajo de lo que puede obtener en un entorno local.
- No tenemos que adivinar la capacidad de nuestro sistema: Con el cloud computing, estos problemas desaparecen ya que en tiempo real podemos escalar hacia arriba o hacia abajo el tamaño del almacenamiento.
- Aumento de la velocidad y la agilidad: En un entorno de cloud computing, los nuevos recursos de TI están a sólo un clic de distancia, lo que significa que reduce el tiempo necesario para poner esos recursos a disposición de sus desarrolladores de semanas a minutos. Esto da lugar a un aumento dramático en agilidad para la organización, puesto que el coste y el tiempo que tarda experimentar y desarrollar es perceptiblemente más bajo.
- Ahorro en manteniendo los centros de datos: La computación en nube nos permite centrarnos en el cliente en lugar del levantamiento pesado de trasiego, apilamiento y servidores.
- Ir global en cuestión de minutos: Esto significa que podemos ofrecer una menor latencia y una mejor experiencia para sus clientes a un costo mínimo ya que con un par de clics podemos subir nuestra aplicación en muchas regiones del globo.

Infraestructura Global

AWS sirve a más de un millón de clientes activos en más de 190 países. Estamos ampliando constantemente la infraestructura global para ayudar a nuestros clientes a lograr una menor latencia y un mayor rendimiento, y asegurar que sus datos residan sólo en la región que especifican. A medida que nuestros clientes crecen sus negocios, AWS continuará proporcionando infraestructura que satisfaga sus requerimientos globales.

La infraestructura de AWS Cloud se basa en Regions and Availability Zones (AZs). Una Región es un lugar físico en el mundo donde tenemos múltiples AZs. Los AZ consisten en uno o más centros de datos discretos, cada uno con redundancia de energía, redes y conectividad, alojados en instalaciones separadas. Estos AZs le ofrecen la capacidad de operar aplicaciones de producción y bases de datos que están más disponibles, tolerantes a fallos y escalables de lo que sería posible desde un único centro de datos. AWS Cloud opera 42 AZ en 16 Regiones geográficas de todo el mundo, con cinco Zonas de Disponibilidad adicionales y dos Regiones más en línea en 2017.

Cada Región Amazónica está diseñada para estar completamente aislada de las demás Regiones Amazónicas. Esto logra la máxima tolerancia a fallas y estabilidad. Cada AZ está aislado, pero los AZ en una Región están conectados a través de enlaces de baja latencia. AWS le proporciona la flexibilidad para colocar instancias y almacenar datos dentro de varias Regiones geográficas, así como a través de múltiples Zonas de Disponibilidad dentro de cada Región. Cada zona de disponibilidad está diseñada como una zona de fallo independiente. Esto significa que las Zonas de Disponibilidad están físicamente separadas dentro de una región metropolitana típica y están ubicadas en llanuras de inundación de menor riesgo (la clasificación específica de la zona de inundación varía según la Región). Además de instalaciones de generación de respaldo discretas (UPS) e instalaciones de generación de respaldo en sitio, cada una de ellas se alimenta a través de diferentes redes de utilidades independientes para reducir aún más los puntos de falla individuales. Los AZ están todos conectados de forma redundante a múltiples proveedores de tránsito de nivel 1.

Seguridad

La seguridad de la nube en AWS es la prioridad más alta. La seguridad en la nube es muy parecida a la seguridad en sus centros de datos locales, sin los costos de mantenimiento de instalaciones y hardware. En la nube, no es necesario administrar servidores físicos o dispositivos de almacenamiento. En su lugar, utiliza herramientas de seguridad basadas en software para supervisar y proteger el flujo de información hacia y desde fuera de sus recursos de la nube.

Una ventaja de AWS Cloud es que le permite escalar e innovar, manteniendo un entorno seguro y pagando sólo por los servicios que utilizamos. Esto significa que puede tener la seguridad que necesitamos a un costo menor que en un entorno local.

AWS Cloud nos permite un modelo de responsabilidad compartida. Mientras AWS gestiona la seguridad de la nube, el usuario es responsable de la seguridad dentro la nube. Esto significa que conservas el control de la seguridad que elijas implementar para proteger tu propio contenido, plataforma, aplicaciones, sistemas y redes de forma diferente a como lo harías en un centro de datos in situ.

AWS nos brinda proporciona herramientas y características específicas de seguridad a través de la seguridad de la red, la gestión de la configuración, el control de acceso y el cifrado de datos.

Por último, los entornos de AWS son continuamente auditados, con certificaciones de los organismos de acreditación a través de las geografías y verticales. En el entorno de AWS, puede aprovechar las herramientas automatizadas para el inventario de activos y los informes de acceso privilegiado.

Ventajas de la seguridad AWS

- **Mantenga sus datos seguros:** La infraestructura de AWS pone en vigor salvaguardas sólidas para ayudar a proteger tu privacidad. Todos los datos se almacenan en centros de datos AWS altamente seguros.
- **Cumplimiento de los requisitos de cumplimiento:** AWS gestiona docenas de programas de cumplimiento en su infraestructura. Esto significa que ya se han completado segmentos de su cumplimiento.
- **Ahorro de dinero:** se reduce los costos utilizando centros de datos AWS. Mantener el más alto nivel de seguridad sin tener que gestionar su propia instalación
- **Escalar rápidamente:** La seguridad varía con el uso de AWS Cloud. No importa el tamaño de su negocio, la infraestructura de AWS está diseñada para mantener sus datos seguros.

Conformidad

AWS Cloud Compliance le permite comprender los sólidos controles que están en marcha en AWS para mantener la seguridad y la protección de datos en la nube. A medida que los sistemas se construyen sobre la infraestructura de AWS Cloud, las responsabilidades de cumplimiento se compartirán. Al vincular las características del servicio centradas en la gobernabilidad y las auditorías con las normas de cumplimiento o auditoría aplicables, los activadores del cumplimiento de AWS se basan en los programas tradicionales. Esto ayuda a los clientes a establecer y operar en un entorno de control de seguridad AWS.

La infraestructura tecnológica que AWS proporciona a sus clientes está diseñada y gestionada en consonancia con las mejores prácticas de seguridad y una variedad de estándares de seguridad de TI. La siguiente es una lista parcial de los programas de aseguramiento con los que AWS cumple:

- SOC 1 / ISAE 3402, SOC 2, SOC 3
- FISMA, DIACAP y FedRAMP
- PCI DSS Nivel 1
- ISO 9001, ISO 27001, ISO 27018

AWS proporciona a los clientes una amplia gama de información sobre su entorno de control de TI en los informes técnicos, informes, certificaciones, acreditaciones y otros certificados de terceros. Más información está disponible en el whitepaper de Riesgo y Cumplimiento⁷ y el Centro de Seguridad de AWS.

Los principales servicios que ofrece AWS los mencionaremos a continuación haciendo hincapié en los dos utilizados en nuestra aplicación: Amazon RDS y Amazon Elastiack Beanstalk.

Computación

Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) es un servicio web que proporciona una capacidad de cálculo segura y redimensionable en la nube. Está diseñado para facilitar la computación a escala web para los desarrolladores.

La interfaz de servicio web simple de Amazon EC2 le permite obtener y configurar la capacidad con un mínimo de fricción. Le proporciona un control completo de sus recursos informáticos y le permite ejecutar en el entorno informático probado de Amazon. Amazon EC2 reduce el tiempo necesario para obtener e iniciar instancias de servidor nuevas (llamadas instancias de Amazon EC2) a minutos, lo que le permite aumentar rápidamente la capacidad, tanto hacia arriba como hacia abajo, a medida que cambian los requisitos de su computadora. Amazon EC2 cambia la economía de la informática al permitirle pagar sólo por la capacidad que realmente usa. Amazon EC2 proporciona a los desarrolladores y administradores del sistema las herramientas para crear aplicaciones resistentes a fallos y aislarse de los escenarios de fallos comunes.

Beneficios

- **Elastic Web-Scale Computing.** Amazon EC2 nos permite aumentar o disminuir la capacidad en cuestión de minutos, no de horas o días. Puede encargar uno, cientos o incluso miles de instancias de servidor simultáneamente. Debido a que todo esto se controla con API de servicio web, su aplicación puede escalarse automáticamente hacia arriba y hacia abajo dependiendo de sus necesidades.
- **Completamente controlado.** Usted tiene el control completo de sus instancias de Amazon EC2. Tiene acceso root a cada uno, y puede interactuar con ellos como lo haría con cualquier máquina. Puede detener su instancia de Amazon EC2 mientras retiene los datos de su partición de arranque y a continuación, reinicie la misma instancia utilizando las API de servicio web. Las instancias se pueden reiniciar remotamente utilizando API de servicio web.
- **Servicios Flexibles de Alojamiento en Nube.** Se Puede elegir entre varios tipos de instancia, sistemas operativos y paquetes de software. Amazon EC2 le permite seleccionar la configuración de memoria, la CPU, el almacenamiento de instancias y el tamaño de partición de arranque que son óptimos para su elección de sistema operativo y aplicación. Por ejemplo, su elección de sistemas operativos incluye numerosas distribuciones de Linux y Microsoft Windows Server.

Integrado

Amazon EC2 está integrado con la mayoría de los servicios de AWS, como Amazon Simple Storage Service (Amazon S3), Amazon Database Relational Database (Amazon RDS) y Amazon Virtual Private Cloud (Amazon VPC) para proporcionar una solución completa y segura para la computación y almacenamiento en la nube a través de una amplia gama de aplicaciones.

Amazon EC2 ofrece un entorno altamente confiable en el que las instancias de reemplazo se pueden encargar de forma rápida y previsible. El servicio se ejecuta dentro de la

infraestructura de redes y centros de datos probados de Amazon. El compromiso de Acuerdo de Nivel de Servicio (SLA) de Amazon EC2 es 99.95% de disponibilidad para cada Región.

Seguro

Amazon EC2 funciona en conjunto con Amazon VPC para proporcionar seguridad y robusta funcionalidad de red para sus recursos de computación.

- Las instancias de cálculo se encuentran en un VPC con un intervalo de direcciones IP especificado. Usted decide qué instancias están expuestas a Internet y que siguen siendo privadas.
- Los grupos de seguridad y las listas de control de acceso a la red (ACL) le permiten controlar el acceso de red entrante y saliente desde y hacia sus instancias.
- Puede conectar su infraestructura de TI existente a los recursos de su VPC utilizando conexiones de red privada virtual (VPN) cifradas IPSec estándar de la industria.
- Puede proporcionar sus recursos de Amazon EC2 como Instancias Dedicadas.¹⁵ Las Instancias Dedicadas son instancias de Amazon EC2 que se ejecutan en hardware dedicado a un solo cliente para aislamiento adicional.
- Puede suministrar sus recursos de Amazon EC2 en Hosts dedicados, que son servidores físicos con capacidad de instancia EC2 totalmente dedicada a su uso. Los hosts dedicados pueden ayudarle a cumplir con los requisitos de cumplimiento y reducir los costos al permitirle utilizar sus licencias de software vinculadas al servidor existentes.

Económico

Amazon EC2 le transmite los beneficios financieros de la escala de Amazon. Usted paga una tarifa muy baja para la capacidad de cálculo que realmente consume. Consulte las Opciones de compra de Instancia de Amazon EC2 para obtener una descripción más detallada.

- Instancias bajo demanda: con instancias de On-Demand, paga la capacidad de cálculo por hora sin compromisos a largo plazo. Puede aumentar o disminuir su capacidad de cálculo en función de las demandas de su aplicación y sólo paga la tarifa por hora especificada para las instancias que utiliza. El uso de instancias de On-Demand le libera de los costos y las complejidades de planificar, comprar y mantener el hardware y transforma lo que normalmente son costos fijos grandes en costos variables mucho más pequeños. Las instancias bajo demanda también eliminan la necesidad de comprar capacidad de "red de seguridad" para manejar los picos de tráfico periódicos.
- Instancias reservadas: las instancias reservadas proporcionan un descuento significativo (hasta un 75%) en comparación con el precio de la instancia On-Demand. Tiene la flexibilidad de cambiar familias, tipos de sistemas operativos y contratos de arrendamiento al mismo tiempo que se beneficia de los precios de Instancia Reservada cuando utiliza Instancias Convertibles Reservadas.
- Instancias puntuales: las Instancias puntuales permiten pujar por la capacidad de almacenamiento de Amazon EC2. Puesto que las instancias Spot suelen estar disponibles con un descuento en comparación con el precio On-Demand, puede reducir significativamente el costo de ejecutar sus aplicaciones, aumentar la capacidad de cálculo y el rendimiento de su aplicación para el mismo presupuesto y habilitar nuevos tipos de aplicaciones de cloud computing.

Amazon Lightsail

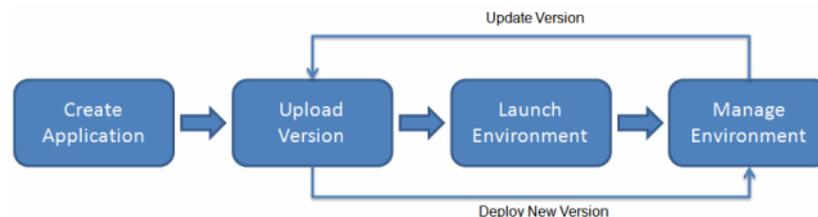
Amazon Lightsail está diseñado para ser la forma más fácil de lanzar y administrar un servidor privado virtual con AWS. Los planes Lightsail incluyen todo lo que necesita para iniciar su proyecto, una máquina virtual, almacenamiento basado en SSD, transferencia de datos, administración de DNS y una dirección IP estática, por un precio bajo y predecible.

AWS Elastic Beanstalk

AWS Elastic Beanstalk es un servicio fácil de usar para implementar y escalar aplicaciones y servicios web desarrollados con Java, .NET, PHP, Node.js, Python, Ruby, Go y Docker en servidores conocidos como Apache, Nginx, Passenger, y Servicios de Internet Information Server (IIS).

Simplemente se puede cargar el código y AWS Elastic Beanstalk se encarga automáticamente del despliegue, desde el aprovisionamiento de capacidad, el equilibrio de carga y la ampliación automática hasta la supervisión de la aplicación. Al mismo tiempo, mantiene el control total sobre los recursos de AWS que alimentan su aplicación y pueden acceder a los recursos subyacentes en cualquier momento.

Para utilizar Elastic Beanstalk, se crea una aplicación, se sube una versión de aplicación en forma de un paquete de código fuente de aplicación. Elastic Beanstalk inicia automáticamente un entorno y crea y configura los recursos de AWS necesarios para ejecutar su código. Después de lanzar su entorno, puede administrar su entorno e implementar nuevas versiones de aplicación. El siguiente diagrama ilustra el flujo de trabajo de Elastic Beanstalk.



Elastic Beanstalk proporciona a los desarrolladores y administradores de sistemas una forma fácil y rápida de implementar y administrar sus aplicaciones sin tener que preocuparse por la infraestructura de AWS. Si ya conoce los recursos AWS que desea utilizar y cómo funcionan, es posible que prefiera AWS CloudFormation para crear sus recursos AWS creando una plantilla. A continuación, puede utilizar esta plantilla para lanzar nuevos recursos de AWS de la misma manera sin tener que volver a personalizar los recursos de AWS. Una vez desplegados sus recursos, puede modificar y actualizar los recursos de AWS de una manera controlada y predecible, proporcionando el mismo tipo de control de versión sobre la infraestructura AWS que ejerce sobre su software.

Las características principales de Amazon BeanStalk

- **Rápido y sencillo de comenzar:** Elastic Beanstalk es la forma más rápida y sencilla de implementar su aplicación en AWS. Simplemente utilice AWS Management Console, un repositorio Git o un entorno de desarrollo integrado (IDE) como Eclipse o Visual Studio para cargar su aplicación, y Elastic Beanstalk gestiona automáticamente los detalles de implementación de aprovisionamiento de capacidad, equilibrio de carga, escalado automático, y monitoreo de la salud de las aplicaciones. En cuestión de

minutos, su aplicación estará lista para usar sin ningún tipo de infraestructura o configuración de recursos de su parte.

- **Productividad del desarrollador:** Elastic Beanstalk provee y opera la infraestructura y administra la pila de aplicaciones (plataforma) por nosotros, para que no tenga que pasar el tiempo o desarrollar la experiencia. También mantendrá actualizada la plataforma subyacente que ejecuta su aplicación con los últimos parches y actualizaciones. Esto nos da oportunidad de podernos centrar en escribir código en lugar de dedicar tiempo a administrar y configurar servidores, bases de datos, equilibradores de carga, cortafuegos y redes.
- **Control completo de recursos:** tenemos la libertad de seleccionar los recursos de AWS, como el tipo de instancia de Amazon EC2, que son óptimos para su aplicación. Además, Elastic Beanstalk le permite "abrir el capó" y mantener el control total sobre los recursos de AWS que alimentan su aplicación.

Elastic Beanstalk

El Elastic Load Balancing (ELB) distribuye automáticamente el tráfico de aplicaciones entrantes a través de múltiples instancias de EC2. Le permite alcanzar mayores niveles de tolerancia a fallos en sus aplicaciones, proporcionando la cantidad necesaria de capacidad de balanceo de carga necesaria para distribuir el tráfico de aplicaciones.

El Balanceo de carga elástico ofrece dos tipos de equilibradores de carga que cuentan con alta disponibilidad, escalado automático y seguridad robusta. Éstos incluyen el equilibrador de carga clásico que enruta el tráfico basado en la información de aplicación o de nivel de red y el equilibrador de carga de aplicaciones que enruta el tráfico basado en información avanzada de nivel de aplicación que incluye el contenido de la solicitud. El balanceador de carga clásico es ideal para un equilibrio de carga simple del tráfico en varias instancias de EC2, mientras que el Load Balanceador de aplicaciones es ideal para aplicaciones que necesitan capacidades avanzadas de enrutamiento, microservicios y arquitecturas basadas en contenedores. Application Load Balancer ofrece la capacidad de enrutar el tráfico a varios servicios o equilibrar la carga a través de varios puertos en la misma instancia de EC2.

AWS Lambda

AWS Lambda le permite ejecutar código sin aprovisionar o administrar servidores. Usted paga solamente por el tiempo de computación que consume - no hay cargo cuando su código no se está ejecutando. Con Lambda, se puede ejecutar código para virtualmente cualquier tipo de aplicación o servicio de backend, todo con administración cero. Simplemente cargue su código, y Lambda se encargará de todo lo necesario para ejecutar y escalar su código con alta disponibilidad. Puede configurar su código para activarlo automáticamente desde otros servicios de AWS o puede llamarlo directamente desde cualquier aplicación web o móvil.

Escalabilidad Automática

Auto Scaling le ayuda a mantener la disponibilidad de la aplicación y le permite escalar su capacidad de Amazon EC2 hacia arriba o hacia abajo automáticamente según las condiciones que defina. Puede utilizar la escala automática para asegurarse de que está ejecutando el número deseado de instancias de Amazon EC2. Auto Scaling también puede aumentar automáticamente el número de instancias de Amazon EC2 durante picos de demanda para mantener el rendimiento y disminuir la capacidad durante las pausas para reducir los costos.

Auto Scaling es adecuado tanto para aplicaciones que tienen patrones de demanda estables y aplicaciones que experimentan variabilidad horaria, diaria o semanal en el uso.

A continuación, hablaremos de las opciones de almacenamiento que ofrece AWS

Almacenamiento

Amazon S3

Amazon Simple Storage Service (Amazon S3) es un almacenamiento de objetos con una sencilla interfaz de servicio web para almacenar y recuperar cualquier cantidad de datos desde cualquier lugar de la web. Está diseñado para ofrecer un 99,999999999% de durabilidad y escala miles de billones de objetos en todo el mundo.

Puede utilizar Amazon S3 como almacenamiento principal para aplicaciones nativas de la nube; como un repositorio masivo, o "lago de datos", para análisis; como objetivo de copia de seguridad y recuperación y recuperación de desastres; y con la informática sin servidor.

Es simple mover grandes volúmenes de datos dentro o fuera de Amazon S3 con las opciones de migración de datos en la nube de Amazon. Una vez que los datos se almacenan en Amazon S3, se pueden clasificar automáticamente en clases de almacenamiento en nube de menor costo y largo plazo como Amazon S3 Standard - Infrequent Access y Amazon Glacier para archivar.

Amazon Elastic File System

Amazon Elastic File System (Amazon EFS) proporciona almacenamiento de archivos simple y escalable para su uso con instancias de Amazon EC2 en AWS Cloud. Amazon EFS es fácil de usar y ofrece una interfaz sencilla que le permite crear y configurar sistemas de archivos rápida y fácilmente. Con Amazon EFS, la capacidad de almacenamiento es elástica, crece y se contrae automáticamente al agregar y quitar archivos, por lo que sus aplicaciones tienen el almacenamiento que necesitan, cuando lo necesitan.

Cuando se instala en instancias de Amazon EC2, un sistema de archivos de Amazon EFS proporciona una interfaz de sistema de archivos estándar y una semántica de acceso al sistema de archivos, lo que le permite integrar perfectamente Amazon EFS con sus aplicaciones y herramientas existentes. Múltiples instancias de EC2 pueden acceder a un sistema de archivos de Amazon EFS al mismo tiempo, lo que permite a Amazon EFS proporcionar un origen de datos común para cargas de trabajo y aplicaciones que se ejecutan en más de una instancia de EC2.

Puede montar los sistemas de archivos de Amazon EFS en sus servidores de centros de datos locales cuando se conecta a su VPC con AWS Direct Connect. Puede montar los sistemas de archivos de Amazon EFS en servidores locales para migrar conjuntos de datos a EFS, habilitar escenarios de ruptura de nube o hacer una copia de seguridad de sus datos locales en EFS.

Amazon EFS está diseñado para ofrecer alta disponibilidad y durabilidad y proporciona rendimiento para un amplio espectro de cargas de trabajo y aplicaciones, incluyendo grandes datos y análisis, flujos de trabajo de procesamiento de medios.

Amazon Aurora

Amazon Aurora es un motor de base de datos relacional compatible con MySQL y PostgreSQL que combina la velocidad y disponibilidad de las bases de datos comerciales de gama alta con la simplicidad y rentabilidad de las bases de datos de código abierto. Amazon Aurora ofrece hasta cinco veces mejor rendimiento que MySQL con la seguridad, disponibilidad y fiabilidad de una base de datos comercial en una décima parte del costo.

Amazon RDS

El servicio Amazon Database Relacional (Amazon RDS) facilita la configuración, operación y escalado de una base de datos relacional en la nube. Proporciona una capacidad rentable y redimensionable mientras administra las tareas de administración de bases de datos que consumen mucho tiempo, liberándote para enfocarte en tus aplicaciones y negocios. Amazon RDS le ofrece seis motores de base de datos conocidos, incluyendo Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle y Microsoft SQL Server.

Beneficios

- **Rápido y fácil de administrar:** Amazon RDS facilita el paso de la concepción del proyecto al despliegue. Utilice la AWS Management Console, la AWS RDS Command Line Interface o las simples llamadas API para acceder a las capacidades de una base de datos relacional de producción inmediata en cuestión de minutos. No hay necesidad de aprovisionamiento de infraestructura, ni necesidad de instalar y mantener software de base de datos.
- **Altamente escalable:** puede escalar los recursos de cálculo y de almacenamiento de su base de datos con sólo unos clics del ratón o una llamada API, a menudo sin tiempo de inactividad. Muchos tipos de motores RDS de Amazon le permiten iniciar una o más réplicas de lectura para descargar el tráfico de lectura de su instancia de base de datos primaria.
- **Disponible y duradero:** Amazon RDS se ejecuta en la misma infraestructura altamente confiable utilizada por otros servicios Web de Amazon. Al proporcionar una instancia de BD Multi-AZ, Amazon RDS replica de forma síncrona los datos en una instancia de reserva en una zona de disponibilidad diferente (AZ). Amazon RDS tiene muchas otras características que mejoran la confiabilidad para bases de datos críticas de producción, incluyendo copias de seguridad automatizadas, instantáneas de base de datos y reemplazo automático de host.
- **Seguro:** Amazon RDS facilita el control del acceso de red a su base de datos. Amazon RDS también le permite ejecutar sus instancias de base de datos en Amazon VPC, lo que le permite aislar sus instancias de base de datos y conectarse a su infraestructura de TI existente a través de una VPN IPsec cifrada estándar del sector. Muchos tipos de motores RDS de Amazon ofrecen cifrado en reposo y encriptación en tránsito.
- **Barato:** Usted paga tarifas muy bajas y solo por los recursos que realmente consume. Además, se beneficia de la opción de precios On-Demand sin compromisos a largo plazo.

Amazon ofrece más de 90 servicios disponibles hoy en día he catalogado los mas populares y los que nos pueden ser útiles a nuestra aplicación haciendo hincapié en Amazon RDS y Elastic Beanstalk los cuales he elegido para subir la aplicación a producción.

2.2.2 COMPARATIVA AMAZON WEB SERVICES VS GOOGLE CLOUD VS MICROSOFT AZURE

Tres principales actores de los servicios de nube de negocios tienen una serie de productos que cubren todo lo que pueda necesitar para sus operaciones en línea. Pero hay diferencias no sólo en los precios, sino también en cómo nombrar y agrupar sus servicios, así que vamos a comparar uno al lado del otro y averiguar lo que ofrecen.

Nos centraremos en los servicios proporcionados por Amazon Web Services (AWS), Google Cloud Platform (GCP) y Microsoft Azure. No cubriremos todos ellos, ni entraremos en muchos detalles sobre la infraestructura de la computación en la nube. Sin embargo, usted estará expuesto a muchos de los productos que puede utilizar, y esperamos familiarizarse con algunos conceptos de nube.

Las compañías icónicas del sector público y privado como: Netflix, AirBNB, Spotify, Expedia, PBS y muchos, muchos más - dependen de los servicios en la nube para apoyar sus operaciones en línea. Esto les permite concentrarse mejor en hacer lo que se conoce, y dejar que muchos de los aspectos técnicos sean atendidos por una infraestructura que ya existe y se está actualizando constantemente. Si tuvieran que implementar la infraestructura física que realmente necesitan para sus operaciones, necesitarían un ejército de técnicos, un montón de presupuesto adicional y tiempo y muchas empresas iniciales nunca superarían estos desafíos técnicos.

Pero esto no se limita a las grandes empresas. Hoy en día, vivimos en un mundo en el que tanto un gran negocio, como dos jóvenes en casa con prácticamente ningún capital inicial, pueden acceder a infraestructura de clase mundial para almacenamiento, computación, administración y más, para hacer el próximo servicio masivo en línea y pagar por solo lo que consumas.

Precios flexibles

Lo que usted paga variará mucho dependiendo de cuánta potencia de procesamiento demanda, cuántas instancias (es decir, cuántos servidores virtuales) despliega y dónde las implementa (más información en la sección "Ubicaciones"). También habrá descuentos significativos para el uso masivo. En cualquier caso, tendrás estas ventajas la mayor parte del tiempo:

- sin costos iniciales
- sin cargos de terminación
- paga solo por lo que usas
- por minuto de facturación

Productos vs Soluciones

Usaremos los términos productos y servicios de manera bastante indistinta; una solución, sin embargo, es un concepto más específico cuando se trata de servicios en la nube. En pocas palabras, una solución es un conjunto de productos pre-configurados orientados a una necesidad muy específica, con abundante documentación, casos de uso y testimonios que le guiarán a través del proceso de adopción de la infraestructura de la nube.

Algunas soluciones típicas son:

- grandes datos - AWS, GCP, Azure
- marketing digital - AWS, Azure
- ecommerce - AWS, GCP, Azure
- juegos - AWS, GCP,
- gobierno - AWS (federal, estatal y local)
- Internet de las cosas (IoT) - AWS, GCP, Azure (mantenimiento predictivo, monitoreo remoto)
- nubes privadas
- hosting revendedor

Computación

Cuando vas a la nube, por otro lado, solo pagas por lo que usas y puedes escalar a miles de nodos de procesamiento en pocos minutos (y soplar tu tarjeta de crédito mientras estás en ella, si no tienes cuidado).

Elastic Compute Cloud (EC2) es el buque insignia de Amazon para la computación escalable a la demanda, compitiendo con Compute Engine de Google y las máquinas virtuales de Azure y conjuntos de escala de máquina virtual. El servicio de Amazon es el más completo, pero como se mencionó, el precio de EC2 puede llegar a ser muy complicado, y lo mismo ocurre con los precios de VM de Azure. La oferta de Google es algo menos flexible, pero los precios son mucho más fáciles de seguir (consulte la sección de precios).

También hay la opción de alquilar procesos informáticos para aplicaciones web y móviles, que pueden ofrecer ahorros significativos cuando se utilizan en lugar de EC2 o Compute Engine si sus aplicaciones se ajustan a las especificaciones de este servicio.

Almacenamiento

El almacenamiento es un pilar clave para los servicios en la nube. En la nube puedes almacenar con la misma facilidad cualquier cosa, desde un grupo de GB a varios PBs (1 petabyte = 1,024 terabytes = 1,048,576 gigabytes). Tenga en cuenta, sin embargo: la aplicación de estas soluciones no es tan trivial, ya que este no es un alojamiento regular para el que sólo necesita un usuario y contraseña para cargar archivos a un FTP. En su lugar, tendrá que interactuar con APIs o programas de terceros, y puede tomar algún tiempo antes de que esté listo para operar su almacenamiento completamente en la nube.

Para almacenar objetos (es decir, prácticamente cualquier cosa), Amazon Simple Storage Service (S3) es el servicio que ha estado funcionando más tiempo, y como tal tiene una extensa documentación, incluyendo webinars gratuitos, toneladas de código de ejemplo y bibliotecas, artículos y tutoriales y foros de discusión muy activos donde los desarrolladores de Amazon proporcionan retroalimentación muy útil sobre una base regular. Por supuesto, Google Cloud Storage y Microsoft Azure Storage proporcionan un servicio tan confiable y robusto, pero los recursos que encontraremos no llegan ni siquiera a los de Amazon.

service		provider	GB/month
Block Storage	w	Rackspace Cloud	\$0.12
Cloud Files	w	Rackspace Cloud	\$0.1
Cloud Storage	w	Google Cloud Platform	\$0.026 (standard) / \$0.02 (DRA ¹)
Data Lake Store	w	Microsoft Azure	\$0.04
Simple Storage Service (S3)	w	Amazon Web Services	\$0.03 (standard) / \$0.0125 (infrequent)
Storage	w	Microsoft Azure	\$0.024 (LRS ²) / \$0.048 (GRS ³) / \$0.061 (RA-GRS ⁴)

Para archivar, también llamado "almacenamiento en frío" (como cuando almacena los objetos que no tienen acceso plano en su mayor parte), disfrute de tarifas más bajas pero también de velocidades de acceso más bajas, de los casos Las características y los precios son muy similares entre los diferentes proveedores, por lo que es muy posible que el tratamiento por qué API que ha implementado en su back-end. Para conocer las especificaciones y los detalles, vea Amazon Glacier, Cloud Storage Nearline de Google y Copia de seguridad de Azure; y compruebe también las soluciones de archivado que ofrecen estos proveedores el Archivo de datos de AWS y Copia de seguridad y Archivo de Azure.

service		provider	GB/month
Cloud Storage Nearline		Google Cloud Platform	\$0.01 (storage) + \$0.01 (retrieval)
Glacier	w	Amazon Web Services	\$0.007
Storage	w	Microsoft Azure	\$0.01 (LRS) / \$0.02 (GRS) / \$0.025 (RA-GRS)

Ubicaciones

Al desplegar sus servicios, es posible que desee elegir un centro de datos que esté cerca de su objetivo principal de usuarios. Por ejemplo, si usted está haciendo bienes raíces o alojamiento minorista en la Costa Oeste de los Estados Unidos, desee desplegar sus servicios allí mismo para minimizar la latencia y proporcionar una mejor experiencia de usuario.

Amazon tiene claramente la cobertura más extensa:



Azure se acerca, con bastante apoyo para Asia:



Google tiene una sólida cobertura en Estados Unidos, pero se queda atrás en Europa y, en particular, en Asia (sólo un lugar en Taiwan), sin cobertura en Sudamérica. Ninguno de ellos puede desplegarse en África.



Bases de datos

Se puede implementar soluciones SQL con el Servicio de Bases de Datos Relacionales de Amazon (RDS) (soportando muchos DBMS), el Cloud SQL de Google y Azure SQL.

También las nuevas soluciones NoSQL están disponibles con DynamoDB de Amazon, Bigtable y Cloud Datastore de Google y el almacenamiento de DocumentDB y Table de Azure, pero si solo es necesario cache. Si ese es el caso, entonces ElastiCache de Amazon y Redis Cache de Azure podría hacerlo.

Podemos ver claramente que Amazon Web services es superior a la competencia ya que ofrece muchos mas servicios a un precio competitivo, también tenemos la ventaja que como es la plataforma más grande de servicios en la nube, no solo tiene una documentación para resolver dudas.

2.3 TEMA 3: IMPORTANCIA DE UNA ALIMENTACIÓN BALANCEADA.

2.3.1 OBESIDAD

La obesidad es una enfermedad crónica multifactorial, fruto de la interacción entre genotipo y ambiente. En países desarrollados como el nuestro, afecta a un gran porcentaje de la población, en todas las edades, condiciones sociales y en ambos sexos. Su prevalencia ha aumentado y continúa aumentando alarmantemente en nuestra sociedad, así como en países de economía en transición, hasta adquirir proporciones epidémicas.

La obesidad aumenta de manera sustancial no tan sólo el riesgo de diabetes y de enfermedad cardiovascular, sino también de ciertos tipos de cáncer y otras enfermedades muy prevalentes, de tal manera que se ha convertido en la segunda causa de mortalidad prematura y evitable después del tabaco. Los pacientes con obesidad mórbida presentan también un aumento de la mortalidad total y sufren una gran estigmatización social y discriminación, ya que este trastorno mórbido muchas veces no se considera una verdadera enfermedad.

El Diagnóstico Clásico: la obesidad se diagnostica siguiendo el estándar de masa corporal, si el porcentaje normal de grasa corporal oscila entre el 12% y 20% en los hombres y entre el 20% y 30% en las mujeres, hablamos de obesidad cuando este porcentaje supera el 25% en los hombres y el 33% en las mujeres. Si bien existen técnicas precisas para determinar el porcentaje de grasa corporal, como la densitometría y los pliegues cutáneos, su coste y/o complejidad limitan su utilización generalizada. Por esta razón, en la práctica clínica y en los estudios epidemiológicos, se utilizan diferentes medidas antropométricas para el diagnóstico del exceso de grasa corporal. De éstas, el IMC, con una elevada correlación con el porcentaje de grasa corporal, es la medida más utilizada

El perímetro de la cintura como factor de riesgo: El PC es la segunda medición obligatoria en la práctica clínica cuando evaluamos a un sujeto obeso, así como en todo su seguimiento. Este perímetro permite estimar la obesidad visceral y el riesgo cardiometabólico, por lo que debe recogerse en la historia clínica. Se determina con una cinta métrica flexible, milimetrada, con el sujeto en bipedestación, sin ropa y relajado. Aparentemente es una determinación sencilla, pero en la práctica requiere un buen adiestramiento por parte del explorador. Aunque no existe un consenso uniforme sobre el punto anatómico de referencia de medida, desde el Consenso SEEDO 2007 ¹ y de acuerdo con la OMS, recomendamos la medida del PC por encima de la cresta iliaca. Para ello se debe localizar el borde superior de la cresta ilíaca en bipedestación, y por encima de este punto, rodear la cintura con la cinta métrica, de manera paralela al plano del suelo, y sin comprimir la piel. La medida se hará al final de una espiración normal, y al igual que la OMS, un perímetro ≥ 102 cm en hombres y ≥ 88 cm en mujeres diagnostica la obesidad abdominal. Sin embargo, debe estimularse que cada población establezca por consenso sus propios puntos de corte de obesidad abdominal, como se recomienda en la definición armonizada de la definición del síndrome metabólico.

Prevalencia de obesidad en España en función del IMC: Muchos de los datos publicados hasta hace unos años sobre prevalencia de obesidad se basaban en las Encuestas Nacionales de Salud, que recogen información sobre peso y talla declarados por el encuestado. Dado que la mayoría de personas tienden a sobrestimar su altura e infraestimar su peso, la prevalencia de obesidad está infraestimada en los datos procedentes de dichas encuestas. Aun así, sabemos que en España la prevalencia de obesidad ha aumentado de forma constante en los últimos 20 años, y que este aumento ha sido más acentuado para los grados más graves de obesidad ($IMC > 40 \text{ kg/m}^2$) 44.

En el año 2003, se comunicó una prevalencia de obesidad en España de 14,5% para el grupo de edad entre 25 y 60 años, siendo mayor en mujeres (15,7%) que en hombres (13,3%) 45. Sin embargo, actualmente disponemos de datos más recientes, con medición directa de peso y talla, sobre la magnitud del problema: los estudios ENRICA 2, Dia@bet.es 46 y Darios 47. En el estudio ENRICA la prevalencia estimada de obesidad, en el periodo comprendido entre 2008 y 2010, fue del 22,9% (24,4% en varones y 21,4% en mujeres). En el Di@bet.es fue del 28,2% (28,9% en varones y 27,5% en mujeres) y en el Darios del 28,0% en varones y 28,3% en mujeres. Los datos más recientes corresponden al periodo 2014-2015, y nos los proporciona el estudio ENPE: del 21,6% entre los adultos entre 25 y 64 años (del 22,8% entre los varones y del 20,5% entre las mujeres) . En todos estos estudios se observó un progresivo aumento de la prevalencia de obesidad con la edad. En general, hasta los 50 años es más prevalente en varones, y a partir de esta edad lo es más en mujeres.

Prevalencia de sobrepeso en España: Las tasas de sobrepeso nos permiten estimar las personas que tienen un estadio previo a la obesidad, y por tanto están en riesgo de desarrollar la enfermedad, y en las que es especialmente importante la puesta en marcha de medidas preventivas. La prevalencia de sobrepeso detectada para varones y mujeres en los cuatro estudios de ámbito nacional fue del 46,4% y 32,5% en el estudio ENRICA 2, 47,5% y 34,0% en el estudio Di@bet.es 46, 50,7% y 35,6% en el estudio Darios, y del 46,5% y 32,1% en el estudio ENPE. Por tanto, en la población adulta española casi la mitad de los varones y un tercio de las mujeres tiene sobrepeso.

Distribución geográfica de la prevalencia de obesidad en España: Existe una importante heterogeneidad en la distribución de la prevalencia de obesidad en España, de forma que clásicamente se ha descrito más alta en el Sur, en concreto en Andalucía, Extremadura, y las Islas Canarias, que en el Norte 2, 46, 47. Sin embargo, los datos más recientes proporcionados por el estudio ENPE sitúan la prevalencia más elevada de obesidad en el Principado de Asturias (25,7%) y en Galicia (24,9%), seguido de Andalucía (24,4%), mientras que las tasas más bajas se observan en Islas Baleares (10,5%), Cataluña (15,5%) y País Vasco (16,8%) .

Factores asociados a la obesidad: sexo, clase social y nivel socio-cultural:

Prácticamente todos los estudios epidemiológicos demuestran consistentemente que la obesidad se asocia a menor nivel socioeconómico y educativo, con lo que su prevalencia se incrementa al reducirse la clase social 49. Los estudios específicos en población laboral española demuestran claramente una mayor prevalencia de obesidad en los trabajadores menos cualificados o manuales, agrupados en la categoría “de cuello azul” versus “cuello blanco” 50. Con un gradiente mucho más pronunciado entre las mujeres: las mujeres de clase social manual presentan un 49% más riesgo de tener sobrepeso y un 96% más riesgo de obesidad que las mujeres de clase social no manual. Mientras, los varones de clase social manual presentan menos riesgo de sobrepeso (18%) y obesidad (12%) frente a los de clase

social manual 51. Por tanto, el género y la clase social son determinantes sociales relevantes de la obesidad en España.

Mortalidad por obesidad en España: Existe una relación directa entre IMC y mortalidad 52,53. Se ha estimado que en el año 2006 se produjeron en España 25.671 muertes atribuibles al exceso de peso, 16.405 en varones y 9.266 en mujeres. La mortalidad atribuible al exceso de peso supuso un 15,8% del total de muertes en varones y un 14,8% en mujeres, pero si nos referimos solo a aquellas causas donde el exceso de peso es factor de riesgo, las cifras se elevan hasta el 31,6% en varones y el 28% en mujeres. La causa más frecuente de mortalidad atribuible a la obesidad fue la enfermedad cardiovascular (58% del total), seguida de la causa tumoral. A nivel individual la mortalidad por exceso de peso tiene una mayor contribución en la DM tipo 2, con cerca de un 70% de los fallecimientos en varones y un 80% en mujeres. El exceso de peso, en su conjunto, es responsable del 54,9% de las muertes en los varones y del 48,6% en las mujeres.

En base a estos datos, la obesidad se convierte en la segunda causa de muerte evitable derivada de hábitos personales, solo superada por el tabaquismo. En pocos años, ante la reducción en el consumo de tabaco, junto al incremento ponderal, la obesidad y el sobrepeso pasarán a convertirse en la primera causa, máxime teniendo en cuenta que el 26,3% de la población infantil tiene exceso de peso y está creciendo de forma geométrica. En este contexto, difícilmente se puede negar que la obesidad sea una enfermedad.

ENFERMEDADES CLÁSICAS ASOCIADAS A LA OBESIDAD.

Visión adipocéntrica del síndrome metabólico (SM): No es hasta que se supera la capacidad del tejido adiposo para almacenar grasa y expandirse, cuando la obesidad produce modificaciones morfológicas y funcionales en el adipocito. En ese momento, el acúmulo de grasa ectópica en órganos distintos al tejido adiposo (tales como el hígado, el músculo, la célula beta pancreática o el miocardio) producirá un efecto tóxico, desarrollando la llamada lipotoxicidad periférica. Este tejido adiposo ectópico y disfuncional será el responsable de la aparición de la resistencia a la insulina y del estado crónico de inflamación sistémica de bajo grado, con un perfil predominante de citoquinas proinflamatorias (como leptina, resistina, IL-6 y TNF α), acompañado de la elevación de ácidos grasos libres, y la disminución en la secreción y actividad de adiponectina.

El SM se define por la asociación de patologías comunes que incluyen la obesidad abdominal, la hipertensión arterial, la dislipemia aterogénica y la hiperglucemia. Considerada la resistencia a la insulina como nexo de unión entre ellas, también se ha sugerido la inclusión dentro de esta definición de patologías como el hígado graso, el SAHS, e incluso patologías psiquiátricas y hasta algunas formas específicas de cáncer.

La alta prevalencia de esta asociación sugiere que estas patologías aparentemente diversas, compartan algunos mecanismos etiopatogénicos comunes, que podrían explicarse desde una visión adipocéntrica del SM. Identificar los mecanismos tempranos que estresan los sistemas homeostáticos, las adaptaciones alostáticas precoces y la vulnerabilidad de los órganos afectados que condicionan diferentes fenotipos dentro del espectro del SM, proporcionará la base para un diagnóstico y un tratamiento individualizados 60.

Obesidad e HTA: El estudio Framingham demostró que el 79% de la hipertensión arterial HTA en varones y el 65% en mujeres fue el resultado directo del exceso de peso. Más allá de la insulinoresistencia y la hiperinsulinemia, la aparición de HTA en la obesidad se relaciona también con otros mecanismos diferentes, como el incremento de la actividad del sistema

nervioso simpático, la producción de angiotensinógeno por parte del tejido adiposo visceral, y el incremento de actividad del sistema renina-angiotensina-aldosterona.

La prevalencia de HTA en individuos obesos es del 25 al 40%, más alta que la de la población general, siendo el riesgo de HTA 5 veces mayor. Según se desprende de los datos del estudio NHANES-III, sabemos también que la prevalencia de hipertensión arterial ajustada por la edad se incrementa progresivamente con el IMC. De forma similar, en el estudio INTERSALT se observó que un exceso de 10 kg de peso suponía un incremento de 3 mm de Hg en la presión arterial sistólica y de 2,3 mm en la diastólica. Estas diferencias en presión arterial suponen un incremento del 12% en el riesgo de enfermedad coronaria y del 24% en el riesgo de ictus 61.

Obesidad y dislipemia: La obesidad, y especialmente la obesidad abdominal, se asocia por sí misma a un perfil lipídico aterogénico: elevación de niveles de triglicéridos, niveles bajos de lipoproteínas de alta densidad (colesterol HDL) y elevación de la fracción de lipoproteínas de baja densidad (colesterol LDL) correspondiente a las partículas pequeñas y densas, siendo el colesterol total generalmente normal. Esta confluencia de alteraciones son el mecanismo mejor establecido entre la obesidad y la enfermedad cardiovascular. Los datos del estudio NHANES- III sobre dislipemia también reflejan la mayor prevalencia de hipercolesterolemia a medida que 18 se incrementa el IMC, especialmente en mujeres. La prevalencia de niveles bajos de HDL (< 40 mg/dl en varones y < 50 mg/dl en mujeres) también se relaciona con el IMC: un descenso de un punto en el IMC implica un cambio de 0,69 mg/dl en el cHDL para mujeres y de 1,1 mg/dl en varones.

Más allá de los parámetros clásicos, también debe destacarse la existencia de formas indirectas de evaluar la dislipemia aterogénica, como son el colesterol no-HDL (valor recomendado, <130 mg/dl) y la determinación de apoproteína B (valor recomendado, <90 mg/dl). Esta última incluye las partículas aterogénicas LDL, IDL, VLDL, quilomicrones remanentes, y Lp(a). También existen cocientes orientativos de riesgo como el cociente cLDL/apoB < 1,3 o el cociente cLDL/triglicéridos > 2,0, sugiriendo ambos mayor número de partículas pequeñas y densas.

Obesidad y diabetes: La obesidad es responsable del 44% de la carga de DM tipo 2, constituyendo, por tanto, el principal factor de riesgo modificable de esta enfermedad 62. Además, en los sujetos con DM tipo 2, la prevalencia de obesidad (según IMC) y obesidad abdominal (según PC) es prácticamente el doble que en la población general. Así, según los datos del estudio Di@bet.es, la prevalencia de obesidad y de obesidad abdominal en los españoles con diabetes conocida alcanza el 50 y el 68%, respectivamente 46.

Se ha establecido un paralelismo entre el incremento de obesidad y el de DM tipo 2 en los últimos 40 años. En el *Nurses Health Study*, que incluyó a 114.281 enfermeras de Estados Unidos con un seguimiento de catorce años, el riesgo más bajo de diabetes se asoció a un IMC inferior a 22 kg/m², incrementándose gradualmente con el aumento de IMC 63. De manera que el riesgo relativo de diabetes ajustado por la edad se incrementó 40 veces en mujeres con un IMC entre 31 y 32,9 kg/m², y hasta 93,2 veces en mujeres con un IMC superior o igual a 35 kg/m². De forma similar, en el *Health Professionals Follow-up Study*, el riesgo más bajo se asoció con un IMC menor a 24 kg/m², y el riesgo relativo de diabetes con un IMC superior o igual a 35 kg/m² fue 42 veces superior.

Obesidad, diabetes mellitus tipo 1 y “doble diabetes”: No hay duda de que el principal mecanismo etiopatogénico de la DM tipo 1 es la destrucción autoinmune de las células β pancreática. Sin embargo, en la última década se ha evidenciado que la resistencia a la insulina asociada al incremento de la obesidad puede jugar también un papel importante en este proceso. Así, la incidencia de DM tipo 1 se ha incrementado entre los jóvenes a un ritmo del 3-4% anual, suponiendo un gran problema de salud pública y la prevalencia de sobrepeso entre los jóvenes con DM tipo 1 es mayor que entre aquellos sin diabetes (22,1% vs. 16,1%, $p < 0,05$).

Por su parte, la ganancia ponderal en sujetos con DM tipo 1 conlleva efectos adversos en forma de mayor necesidad de insulina y mayor riesgo de enfermedad cardiovascular, con incrementos del grosor del complejo íntima media carotídeo en el seguimiento largo plazo en comparación a los no obesos.

Obesidad y síndrome de apnea hipoapnea del sueño (SAHS): La obesidad es el factor de riesgo más importante para el desarrollo del SAHS, trastorno crónico y progresivo asociado no solo con una mayor somnolencia diurna, sino también con un mayor riesgo hipertensión, enfermedad cardiovascular, trastornos metabólicos y accidentes de vehículos de motor. Por su parte, dos tercios de los pacientes con SAHS son obesos. Hipótesis recientes sugieren una potencial contribución del SAHS en la perpetuación de la obesidad. En primer lugar, la somnolencia diurna puede traducirse en una disminución de la actividad física, con la consiguiente disminución del consumo metabólico. En este sentido, existe una estrecha correlación entre el índice de apneas-hipoapneas y el grado de actividad diurna. También se ha demostrado que la reducción del sueño delta por un lado, y los episodios repetidos de activación simpática consecuencia de las apneas por otro, puede producir un estado de insulinoresistencia, asociado a la mayor propensión a la diabetes de estos sujetos .

Obesidad y enfermedad psiquiátrica: En un estudio epidemiológico transversal, con una muestra de 9.125 adultos representativa de la población americana, realizado mediante entrevista personal y obteniendo una tasa de respuesta del 70.9%, se observó como la obesidad se asociaba con un 25% más de posibilidades de trastornos del estado de ánimo y ansiedad, aunque factores sociales y culturales juegan un importante papel y pueden moderar o mediar esta asociación . Por su parte, un estudio longitudinal a 8 años realizado en población infantil entre 9 y 16 años, asocia únicamente la obesidad crónica con la presencia de psicopatología.

Obesidad e hígado graso: La enfermedad hepática no alcohólica por depósito de grasa (EHNA) comprende un espectro de trastornos del hígado desde la esteatosis simple hasta la esteatohepatitis y cirrosis. En todos ellos la lipotoxicidad hepatocelular es el factor patogénico fundamental, y la obesidad está claramente implicada en la progresión de la enfermedad. La prevalencia de EHNA entre las personas con obesidad es del 76%, y en pacientes con obesidad mórbida, de casi el 100%. La pérdida de peso mantenida en el tiempo es la mejor estrategia terapéutica para revertir la EHNA. En este sentido, tras un año de consejo dietético intensivo, se objetivó una mejoría histológica asociada a la mayor pérdida de peso en 9 de 15 pacientes con biopsia previa de esteatohepatitis no alcohólica. Con la pérdida ponderal conseguida tras cirugía bariátrica se resuelve la enfermedad grasa hepática en casi la totalidad de sujetos obesos intervenidos.

Obesidad y osteoartritis: Conjuntamente con la edad, la obesidad es el factor más claramente implicado en el desarrollo de la artrosis, tanto de cadera como de rodilla, pero también en articulaciones que no son de carga como las manos. Esta asociación se observa

tanto en varones como en mujeres. Sobre el cartílago articular no solo influye el efecto negativo directo del exceso de peso, sino también la acción lesiva directa de las citoquinas provenientes del tejido adiposo. Así, el porcentaje de artrosis es mayor entre aquellos sujetos obesos que además presentan otras comorbilidades como la DM tipo 2, la hipertensión arterial o la dislipemia. Además, la progresión radiológica de la artrosis es más rápida en los pacientes con obesidad. La pérdida ponderal moderada, de al menos un 5% en un periodo de 20 semanas, disminuye de forma significativa la discapacidad física y el dolor de los pacientes con osteoartritis de rodilla.

2.3.2 ALIMENTACION BALANCEADA

Para combatir la obesidad es necesario llevar un plan de alimentación y plan de actividad física.

Estilo de vida saludable: La prevención y el tratamiento de la obesidad deben enfocarse de forma integral hacia la consecución de un estilo de vida saludable, lo que incluye dos pilares fundamentales como son la alimentación y el ejercicio físico. Y lo más importante, los cambios deben no solo llevarse a cabo, si no también mantenerse a largo plazo. Sin embargo, un “estilo de vida saludable” implica a otros determinantes más allá de la salud (sociales, geopolíticos, ambientales y en definitiva “ecológicos”), constituyendo el individuo una parte de un ecosistema que condiciona gran parte de sus comportamientos y desarrollos. El concepto de salud psico-bio-social es, por lo tanto, tremendamente amplio, y los factores determinantes de la misma engloban desde los aspectos más biológicos, como los caracteres genéticos, a otros socioeconómicos y culturales que, en conjunto, determinan la situación de salud de los individuos. Es por ello que la existencia de patrones de salud asociados a un menor riesgo de obesidad centra este punto. Así, por ejemplo, el patrón de vida “mediterránea” se ha asociado a la mejoría en algunos determinantes fundamentales de la salud, como los eventos cardiovasculares, y una menor mortalidad general.

¿Cómo alimentarse?: El patrón de alimentación, dentro de los estilos de vida saludables, incluye siempre un mayor consumo de verduras y frutas, seguido, aunque con menor grado de evidencia, de cereales integrales, lácteos bajos en grasa, pescado, legumbres y frutos secos. La disminución en la ingesta de carnes, incluyendo carnes procesadas, y de alimentos con azúcar, en especial las bebidas, también ha sido identificada como características de los patrones de alimentación saludables.

Alimentación saludable: El tratamiento dietético de la obesidad incluye diversas medidas que, a través de la modificación de la alimentación, buscan revertir el exceso de peso y masa grasa que la caracterizan. Estas modificaciones pueden ser de tipo cuantitativo (reduciendo las cantidades de alimentos consumidas de forma habitual o raciones, y por tanto, el aporte energético), de tipo cualitativo (variando la proporción de diferentes nutrientes), y de tipo mixto, que combina modificaciones cuantitativas y cualitativas 1. Este cambio en la alimentación debe ir dirigido a la disminución de la grasa corporal, preservando la masa magra a largo plazo, previniendo la ganancia de peso, y englobarse en un cambio de hábitos

que consiga disminuir los factores de riesgo cardiovascular y que mejore otras comorbilidades, aumentando la capacidad funcional y la calidad de vida del paciente.

La “alimentación ideal”: Se han establecido “modelos de dieta” que tratan de adaptarse a estas condiciones y características de la “alimentación ideal”. Ninguno las cumple totalmente, pero sí son útiles como referentes para evaluar sus diferentes características, limitaciones y oportunidades para individualizar el enfoque en cada paciente. No existe una clara unanimidad respecto a la “alimentación ideal” por no disponer de datos en la literatura que nos permitan claramente contrastar unas con otras.

Podría diferenciarse un efecto a corto plazo, de 3-12 meses, centrado en la pérdida de peso, y un efecto a largo plazo, durante el resto de la vida, enfocado en evitar la ganancia del peso perdido. En los diferentes modelos de alimentación se han evidenciado algunas diferencias a corto plazo que no se mantienen en el tiempo.

La dieta hipocalórica “clásica”: Es la recomendada en la mayoría de guías y consensos 106. Establece un déficit de 500 a 1.000 calorías cada día respecto a la ingesta habitual del paciente, lo que supone un aporte energético de unas 1.000 a 1.500 calorías para las mujeres, y entre 1.200 y 1.800 calorías para los varones. Esta reducción energética supone una pérdida promedio de 0,5-1 kg/semana, y aproximadamente entre un 5-10% del peso inicial tras 6 meses de tratamiento. Para perder 1 kilo de grasa deberemos realizar un balance negativo de 7.200 calorías.

Respecto a la distribución de nutrientes de la dieta, la mayoría de las guías han recomendado una dieta hipocalórica equilibrada: 45-55% de hidratos de carbono, 15-25% de proteínas, 25-35% de grasas totales (7% de grasas saturadas, 15-20% de grasas monoinsaturadas, 7% de grasas poliinsaturadas, <2% de ácidos grasos trans) y 20-40 gramos de fibra 1. La reducción de la grasa en el contexto de una dieta hipocalórica es un método práctico y eficaz para reducir calorías y conseguir pérdida de peso. Su eficacia en la pérdida de peso está demostrada claramente en la literatura (5-10% de pérdida de peso entre los 3 y 12 meses de seguimiento, con disminución de la grasa abdominal y una reducción del PC, con reducción de los niveles de presión arterial, colesterol LDL y triglicéridos. Su principal limitación es su escasa adherencia y elevada tasa de abandonos, que mejora cuando esta dieta se incluye en un programa global de cambio de hábitos con ejercicio y apoyo psicológico.

La dieta mediterránea: Es el modelo con preferencia por ser el que mejor representa en la práctica clínica este enfoque equilibrado dentro de una estructura de hábitos saludable. La dieta mediterránea “hipocalórica” se basa en una ingesta baja de ácidos grasos saturados, *trans* y azúcares añadidos, un alto consumo de fibra y ácidos grasos monoinsaturados. Es importante resaltar que, para completar la restricción de los hidratos de carbono en este modelo de dieta hipocalórica, junto a la disminución de los azúcares simples deben limitarse también los alimentos energéticos de alta carga glucémica, como el pan, las patatas, el arroz, la pasta y cereales refinados. Estos alimentos pueden sustituirse por un mayor consumo de frutas, hortalizas, cereales integrales y frutos secos. Los beneficios de la dieta mediterránea están claramente establecidos en la literatura en diferentes aspectos de la salud como por

ejemplo la mortalidad, el riesgo cardiovascular o la incidencia de cáncer, si bien su efecto en el peso está en discusión.

El ejemplo de la “dieta del plato”: El planteamiento de la pauta alimentaria con grupos de alimentos constituye una estrategia de elaboración de las recomendaciones nutricionales de forma práctica. Los métodos de distribución del “plato”, con grupos de alimentos (lácteos, frutas, verduras, proteínas y carbohidratos), permiten la adaptación de las recomendaciones de energía y reparto equilibrado de macronutrientes a una alimentación saludable.

Dietas modificadas en macronutrientes: En referencia a la distribución de macronutrientes de la dieta, y ante la ausencia de ensayos clínicos bien diseñados que permitan establecer claramente las ventajas de un tipo de dieta frente a otra, la mayoría de las guías internacionales siguen las recomendaciones generales de una dieta hipocalórica equilibrada. Sin embargo, cada vez surgen más datos que obligan a reconsiderar otros enfoques dietéticos. Los beneficios potenciales de estos enfoques alternativos (dietas cetogénicas, dietas hiperproteicas, dietas de bajo índice glucémico, etc...), solo durante periodos de intervención concretos y en determinados pacientes, aún deben ser estudiados en profundidad 106. Sin embargo, su generalización, la ausencia de un control médico estricto o la presencia de intereses comerciales en determinadas ocasiones, hacen que su uso no sea recomendado en las guías salvo casos individualizados y con seguimiento clínico correcto. Ello justifica la necesidad de estudios científicos que justifiquen la recomendación de un cambio en la proporción de macronutrientes en el tratamiento de la obesidad, y que nos permitan en el futuro la individualización de las recomendaciones dietéticas.

Dietas cetogénicas y bajas en hidratos de carbono: Estos modelos de dieta establecen una reducción energética a expensas fundamentalmente de la disminución de calorías procedentes de los hidratos de carbono, aunque uno de los principales problemas es que no existe uniformidad absoluta en la literatura sobre lo que se entiende por una dieta baja en hidratos de carbono. Generalmente hace referencia a una disminución del aporte de hidratos de carbono a menos de 20 a 60 g/día (menos del 20% del aporte calórico total), junto con un incremento proporcional en el aporte de grasas o proteínas para compensarlo 110. La cetosis se establece con una restricción de carbohidratos a menos de 20 g al día, lo que supone un aumento considerable del aporte de proteínas o grasas para poder suplir el aporte calórico determinado. La conciliación de este esquema con la limitación universalmente aceptada de mantener 7% de ingesta de grasa saturadas hace a este modelo prácticamente imposible de alcanzar con alimentos naturales, y queda limitado a sustitución de alimentos en periodos de intervención a corto plazo.

Las dietas populares de este perfil, tipo Atkins, con alimentos naturales presentan características nutricionales, metabólicas y de efectos adversos de importancia en su evaluación

a largo plazo, como el mayor consumo de grasas de origen animal asociado a mayor mortalidad fundamentalmente de origen cardiovascular. Cuando la intervención se realiza a través de dietas artificiales de sustitución cetogénicas, con un patrón nutricional bajo en

grasas saturadas y con suplementación en vitaminas, minerales y algunos nutrientes funcionales moduladores de la respuesta metabólica, se aprecia un beneficio en la pérdida ponderal a corto y medio plazo, si bien aún carecemos de revisiones sistemáticas y metanálisis que nos proporcionen suficiente base científica para aconsejar su uso de forma generalizada 110. Los efectos secundarios descritos (estreñimiento, cefalea, halitosis y calambres musculares, déficits vitamínicos o minerales) hacen necesario un adecuado control médico para valorar la necesidad de suplementos vitamínicos-minerales, fibra, etc, así como disponer de estudios de seguridad con este tipo de dietas 111.

En general, según los estudios realizados, estas dietas producen una pérdida de peso mayor durante los primeros 6 meses, con una mejoría del perfil lipídico (colesterol HDL y triglicéridos), reducción de la resistencia a la insulina, pero esta diferencia se pierde a partir de los 12 meses, probablemente por la falta de adherencia en mantener una dieta con muy bajo contenido en hidratos de carbono. Cuando se comparan dietas isocalóricas con diferentes proporciones de macronutrientes, no parece que las pérdidas de peso se relacionen con un mayor o menor porcentaje de los macronutrientes que la componen, sino más bien parece que guardan relación con en el grado de adherencia del paciente a la dieta asignada. El papel de la cetosis en el control de la ingesta y sus efectos metabólicos necesitan estudios más amplios. Está también en estudio la readaptación a una dieta de mantenimiento a largo plazo.

Dietas hiperproteicas: Otra de las modificaciones en la proporción de macronutrientes de la dieta orientadas a inducir un mayor efecto sobre la pérdida de peso es el incremento del contenido de proteínas de la misma. Las proteínas podrían inducir un efecto saciante mayor que otros principios inmediatos, y una menor pérdida de masa magra durante la fase de inducción de la dieta hipocalórica. Más que la proporción de proteínas, entre el 20-30% del aporte calórico total, lo más importante desde el punto de vista nutricional es la cantidad absoluta de proteínas ingeridas (g proteínas / kg de peso). En condiciones normales el aporte fisiológico se considera entre 0,8-1,2g de proteínas/kg/día, mientras que las dietas hiperproteicas aportan cantidades de proteínas superiores a 1,2g/kg de peso/día o un aporte mínimo global de 90g/día de proteínas 106.

Estas dietas hiperproteicas hipocalóricas pueden representar una mayor o menor restricción de carbohidratos, y por tanto, una mayor o menor abundancia de lípidos, lo cual hace difícil la comparación entre los estudios. Fuera del marco de los ensayos clínicos, existen dietas populares que pueden considerarse hiperproteicas como la dieta Atkins, la dieta de la Zona (40- 30-30 entre hidratos de carbono, proteínas y grasas respectivamente), y la dieta Dukan.

La mayor parte de estudios suelen describir una mayor pérdida de peso entre los 3 y 6 meses iniciales, diferencia que desaparece a largo plazo. Parecen producir una mayor reducción de los niveles de triglicéridos y un aumento en los niveles de vitamina B12. El principal problema que plantean es el incremento del consumo crónico de proteínas de origen animal, asociado a un aumento de mortalidad total y cardiovascular, probablemente relacionados con efectos secundarios a largo plazo como alteraciones en la microbiota intestinal o el incremento de potenciales agentes carcinogénicos en el tubo digestivo 112.

En la actualidad ya se establece en alguna guía clínica la posibilidad de que las dietas hiperproteicas moderadas o bajas en grasas puedan ser una alternativa en el tratamiento de la obesidad a corto plazo 113.

¿Dieta hiperproteica para mantener el peso perdido a largo plazo?: Tras la pérdida de peso inducida por una dieta hipocalórica, una dieta con mayor contenido proteico y un menor índice glucémico podría tener un papel en el mantenimiento a largo plazo del peso perdido, al facilitar la pérdida de grasa preservando la masa muscular, hecho fundamental para el mantenimiento del gasto metabólico basal 114. Sin embargo, las dificultades de autogestión por parte del paciente, el incremento de proteínas de origen animal y los posibles déficits nutricionales asociados, limitan sus posibilidades de recomendación a nivel poblacional.

Dietas bajas en grasas: Las grasas en alimentación, por su elevado valor energético, se han considerado clásicamente como unos de los factores responsables de la etiopatogenia de la obesidad. Se considera una dieta baja en grasa aquella que contiene menos de un 20% de grasa, y como muy baja en grasa la que contiene menos del 10% 115. El aporte en proteínas suele ser estable (15-20%), lo que supone un incremento proporcional en hidratos de carbono, que las convierte en dietas muy ricas en este principio inmediato y también muy ricas en fibra.

La eficacia en la pérdida de peso, a igualdad de aporte calórico, es similar a la dieta hipocalórica convencional tanto a corto como a largo plazo 107, 116. Producen una reducción de los niveles de colesterol LDL y de la presión arterial, pero en cambio pueden originar un incremento de los triglicéridos que se acompaña con una disminución del colesterol HDL. Según la fuente de carbohidratos se pueden producir diferentes efectos sobre la sensibilidad a la insulina.

Realmente, aunque no existe una limitación teórica de la cantidad de alimentos que se pueden ingerir de los grupos recomendados, la disminución de peso que se observa con este tipo de dietas se ha relacionado con el efecto saciante que tiene la gran cantidad de alimentos que se tienen que ingerir con gran contenido en fibra y palatabilidad limitada, lo que conduce finalmente a una disminución del aporte energético. Son dietas sin efectos secundarios graves, salvo distensión abdominal, molestias digestivas y flatulencia o dificultades de adherencia que mejoran incrementando el contenido de grasas (<20%). Algunas dietas populares de este perfil, como la dieta Ornish o la dieta Pritikin, pueden disminuir la absorción de cinc, calcio y hierro por el alto aporte de fibra que interfiere en su absorción. Según el aporte de alimentos de origen animal también pueden asociarse al déficit en vitamina E y B12.

Dietas con modificación de hidratos de carbono: Existen dietas que tratan de modificar no el porcentaje global de carbohidratos sino las características de los mismos, en cuanto a su índice glucémico (incremento en la glucemia tras la ingesta de 50 g alimento/ incremento en la glucemia tras tomar 50 g de pan blanco o de glucosa x 100) o la carga glucémica (índice glucémico del alimento x cantidad total de carbohidratos de la ración habitual de dicho alimento). Esta propuesta se basa en el efecto de los alimentos con bajo índice (< 55) y carga (< 10) glucémica sobre la glucemia postprandial, que podría incrementar la sensación de saciedad. Sin embargo, en los estudios realizados, a igualdad de contenido calórico total, esta modificación no induce una mayor pérdida de peso durante la realización de una dieta hipocalórica, aunque sí puede tener un papel en el mantenimiento a largo plazo del peso perdido 105, 106. Dietas populares, como la dieta Montignac o la dieta de la Zona se podría incluir en este grupo.

Aunque no parece que conlleve efectos adversos de importancia, los datos de los efectos metabólicos sobre los parámetros lipídicos o de riesgo cardiovascular son variables y heterogéneos, sin que se pueda concluir un claro beneficio en el problema de la obesidad. Ni siquiera en pacientes con trastornos del metabolismo de carbohidratos y DM tipo 2 se evidencian claros beneficios de estas estrategias de modificación de los carbohidratos 117.

Dietas “milagro”: Existen diversas dietas heterodoxas “populares”, con mayor o menor fundamento, que ofrecen múltiples propuestas nutricionales diferentes a los esquemas dietéticos recomendados en las guías clínicas. Muchas de ellas, basadas en modificaciones dietéticas totalmente desequilibradas, como las “mono-dietas” (solo manzana, piña o jarabe de agave) son claramente perjudiciales para la salud y no están respaldadas por ningún tipo de estudio clínico que avale su uso en ningún tipo de paciente. Es fácil deducir que inducirán un déficit energético que no solo produce la pérdida de peso, sino también de nutrientes con elevado riesgo de desnutrición proteica, múltiples deficiencias en vitaminas y minerales.

Dentro de este epígrafe se engloban también las dietas hipocalóricas desequilibradas (dieta “toma la mitad”, dieta de la Clínica Mayo, dieta Gourmet o dieta Cero) y las dietas disociativas (Dieta de Hay o disociada, régimen de Shelton, dieta de Montignac, la antidieta o la dieta de Raffaella Carrà) producen confusión al venir dirigidas desde personalidades populares y otras personas de origen científico que se consideran autorizadas para transmitir consejos y prescribir dietas. Otras dietas están basadas en multitud de mitos y prejuicios (“sopa quema-grasa”, cronodieta, grupo Sanguíneo, test de Alcat, dieta de las frutas) difundidas por diversos canales de comunicación y muy presentes en la redes sociales, pero sin ningún estudio científico que las avale.

Las propuestas dietéticas excluyentes, pobres en grasa, pobre en hidratos de carbono e hiperproteicas o hipergrasas, deben ser consideradas como potencialmente “peligrosas” si no son controladas por profesionales sanitarios médicos adecuadamente cualificados para el manejo de sus contraindicaciones y principales efectos secundarios.

Peligros de una dieta inadecuada: Algunos de los riesgos de una dieta inadecuada pueden ser originar situaciones de desnutrición o déficit de micronutrientes, empeorar el riesgo cardiovascular de los pacientes, favorecer el desarrollo de trastornos de la conducta alimentaria, transmitir conceptos nutricionales erróneos, fomentar el sentimiento de frustración, etc.

Alimentos funcionales: El incremento en la incidencia de la obesidad asociada a la enorme presión realizada sobre los consumidores, ofrece una oportunidad única para la industria alimentaria para aportar alimentos que ayuden a la prevención y/o al tratamiento de estas patologías. Estos alimentos funcionales tratan de establecer estrategias para el control del peso basadas en la inhibición de la ingesta o limitación de la biodisponibilidad de nutrientes, producción de un descenso en el contenido calórico de los alimentos, estímulo del gasto energético (termogénesis), etc.

Sin embargo, el uso de muchas de estas sustancias no está sustentado por evidencias científicas firmes sobre su eficacia, y lo que puede ser más preocupante, sobre su seguridad a largo plazo. Las políticas sanitarias deben advertir sobre los posibles riesgos y dudosos beneficios de algunos de estos suplementos 118. Aunque existen principios activos con

características funcionales prometedoras (ácido linoleico conjugado o CLA, piruvato, o *irvingia gabonensis*), la experiencia de uso es aún muy limitada. La SEEDO no promueve la pérdida de peso a través del uso de sustancias tales como los extractos de efedra, el ginseng, el cromo y la L-Carnitina, la *Garcinia Cambogia*, el azafrán, el chitosano, el “mango africano” o la “cetona de frambuesas” entre otras.

MARCO PRÁCTICO

MARCO PRÁCTICO.

3.1. INTRODUCCIÓN

La aplicación constituye un sistema de apoyo para el control alimenticio de una persona llevando un control de cada comida que ha registrado en el sistema tomando en cuenta los alimentos que ha comido y contabilizando las calorías de esa comida (desayuno, comida, cena o merienda). El sistema registrará la ingesta calórica y podrá ser consultado por semana y será mostrado por día la media en calorías ingeridas y si el paciente es diabético podrá también introducir sus niveles de glucosa antes de una comida.

Esta aplicación va dirigida a los nutricionistas quienes quieran ofrecer a sus pacientes esta aplicación, que a través de ella podrá concertar sus citas con sus pacientes y llevar un control de lo que el paciente coma así podrá hacer recomendaciones en su dieta para lograr el balance correcto para la persona.

Para poder disfrutar de la aplicación primero habrá que registrarse en la aplicación que puede ser a través del formulario de la aplicación o a través de las redes sociales ofrecidas: Facebook, Twitter, Instagram. Una vez registrado existen dos menús como doctor o como paciente, como paciente tendrá un menú de 4 opciones:

- Datos Personales: donde podrá consultar y modificar sus datos personales
- Citas: Donde podrá concertar sus citas con su doctor asignado.
- Comidas: donde podrá registrar una comida realizada.
- Progreso donde podrá consultar por semana un gráfico de sus calorías

Si entra como doctor de inicio se mostrará los pacientes que tiene asignados con toda su información necesaria, donde podrá acceder a su calendario y concertar sus citas con él o ver el progreso de su paciente, consultando por semana el nivel de calorías consumidas.

Para el desarrollo de esta aplicación fue necesario los datos de la base de datos BEDCA anteriormente mencionada, el backend de la aplicación también tiene una serie de dependencias para su correcto funcionamiento ya que como hemos hablado antes de Django es una plataforma a la cual se le puede ampliar funcionalidad instalando paquetes e integrándolas a la aplicación

- awsclicli==1.11.132
- beautifulsoup4==4.6.0
- blessed==1.14.2
- botocore==1.5.84
- cement==2.8.2
- certifi==2017.4.17
- chardet==3.0.4
- colorama==0.3.7
- cssutils==1.0.2
- Django==1.9.6
- django-annoying==0.10.3
- django-bootstrap-form==3.2.1
- django-bootstrap-form-horizontal==0.0.4
- django-bootstrap-toolkit==2.15.0
- django-bootstrap3==8.2.3
- django-bootstrap3-datetimepicker-2==2.4.2
- django-crispy-forms==1.6.1
- django-datetime-widget==0.9.3
- django-inlineness==0.1.2
- django-libs==1.67.12
- django-mptt==0.8.7
- django-polymorphic==1.0.2
- django-scheduler==0.8.5
- django-sslserver==0.19
- docker-py==1.7.2
- dockerpty==0.4.1
- docopt==0.6.2
- docutils==0.13.1
- icalendar==3.11.5
- idna==2.5
- jmespath==0.9.3
- mock==2.0.0

- packaging==16.8
- pathspec==0.5.0
- pbr==3.1.1
- PyJWT==1.5.2
- PyMySQL==0.7.11
- pynliner3==0.6
- pyparsing==2.2.0
- python-dateutil==2.6.0
- python-openid==2.2.5
- pytz==2017.2
- requests==2.9.1
- requests-oauthlib==0.8.0
- semantic-version==2.5.0
- six==1.10.0
- social-auth-app-django==1.2.0
- social-auth-core==1.4.0
- tabulate==0.7.5
- termcolor==1.1.0
- Unidecode==0.4.20
- wcwidth==0.1.7

Para realizar el front-end de la aplicación se ha utilizado las siguientes tecnologías

- HTML5
- jQuery es una biblioteca JavaScript rápida, pequeña y rica en funciones. Hace cosas como el desplazamiento de documentos HTML y la manipulación, manejo de eventos, animación y Ajax mucho más simple con una API fácil de usar que funciona a través de una multitud de navegadores
- Hojas de estilo
 - Bootstrap: Bootstrap es un conjunto de herramientas de código abierto para desarrollar con HTML, CSS y JS personalizado
- Imágenes
- Font Awesome un framework de iconos para web

- Full Calendar: un framework en javascript para facilitar el desarrollo del calendario de la aplicación.
- D3.js es una biblioteca JavaScript para manipular documentos basados en datos. D3 le ayuda a llevar datos a la vida usando HTML, SVG y CSS. El énfasis de D3 en los estándares web le ofrece las capacidades completas de los navegadores modernos sin tener que atarse a un framework propietario, combinando componentes de visualización potentes y un enfoque basado en datos para la manipulación DOM.

3.2 METODOLOGÍA

La metodología seguida en el proyecto ha sido la metodología de desarrollo en cascada, también llamado modelo en cascada. Esta metodología se caracteriza por seguir un orden riguroso en las distintas etapas en las que se divide el desarrollo del software del proyecto. Cabe destacar que esencial seguir el orden de las etapas y no empezar la siguiente sin que la actual etapa se haya realizado. Las etapas seguidas son las siguientes:

- Análisis de requisitos.
- Diseño del sistema.
- Diseño del programa.
- Codificación.
- Pruebas.
- Subida a producción.
- Mantenimiento.

3.2.1 RECOGIDA DE REQUISITOS

Los requisitos a cubrir por la aplicación son:

- Registrarse en la aplicación: el usuario podrá registrarse en la aplicación utilizando el formulario básico.
- Inicio de Sesión: el usuario podrá hacer inicio de sesión a través de sus redes sociales o con usuario y contraseña del método tradicional.
- Cambiar contraseña: el usuario siempre y cuando sea haya registrado a través del formulario de la página podrá recuperar su contraseña.
- Modificar datos personales: el usuario podrá modificar sus datos personales.
- Concertar una cita: el usuario de la aplicación podrá concertar una cita con su paciente/doctor.
- Registrar una comida: el usuario podrá escoger entre registrar desayuno, almuerzo, cena o merienda para luego escoger que alimentos y cuantos a gramos ha comido de cada alimento.
- Registrar el nivel de glucosa: el usuario si es diabético podrá registrar el nivel de glucosa antes de cada comida que haga.
- Consultar su ingesta calórica: el usuario será capaz de consultar la cantidad de calorías registradas por semana que serán reflejadas en un gráfico.

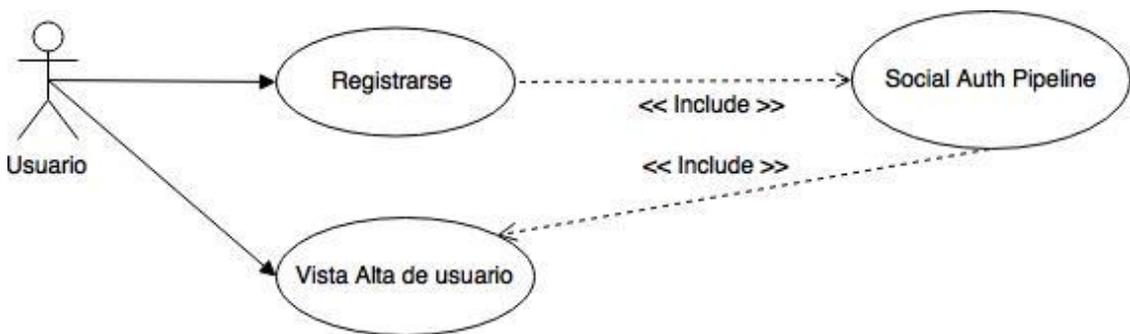
- Consultar sus niveles de glucosa: el usuario será capaz de consultar los niveles de glucosa por semana que serán reflejadas en un gráfico

3.2.2 CASOS DE USO

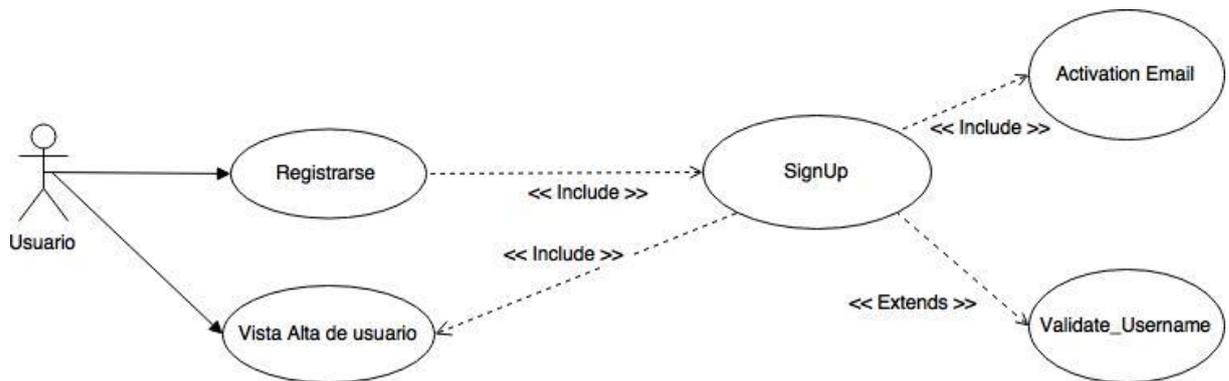
Para terminar de definir correctamente se describen a continuación los diagramas de caso de uso que nos permiten hacernos una idea de la funcionalidad de las distintas partes que componen la aplicación.

1. Registrarse en la aplicación:

El usuario elige por donde registrarse si por redes sociales o por el formulario de la aplicación una prueba con las irregularidades detectadas.



A través de redes sociales.

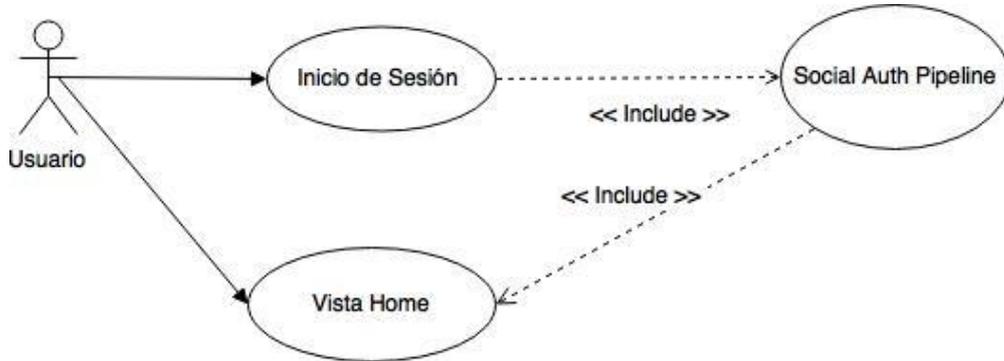


A través del formulario de la aplicación

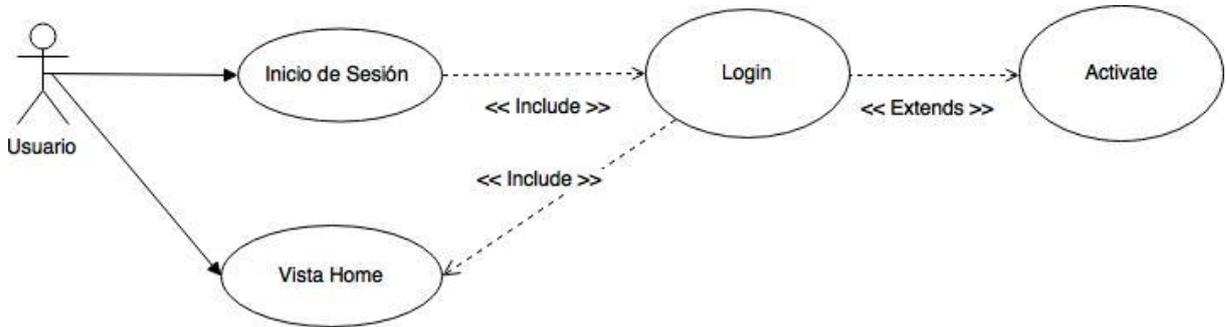
2. Inicio de Sesión:

El usuario para entrar a la aplicación puede elegir dos opciones dependiendo de cómo ha sido dado de alta:

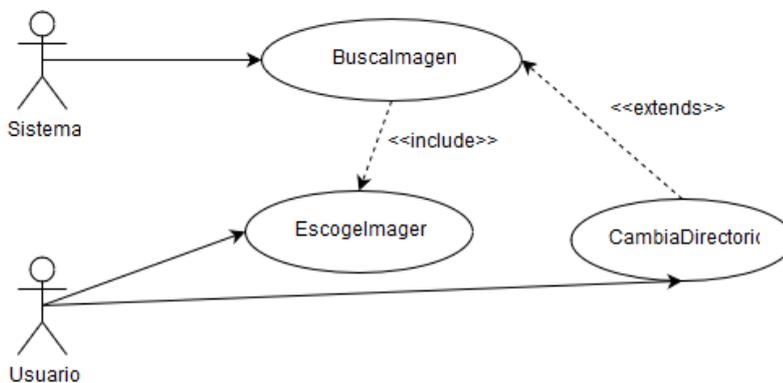
- Por redes sociales



- Por la aplicación :

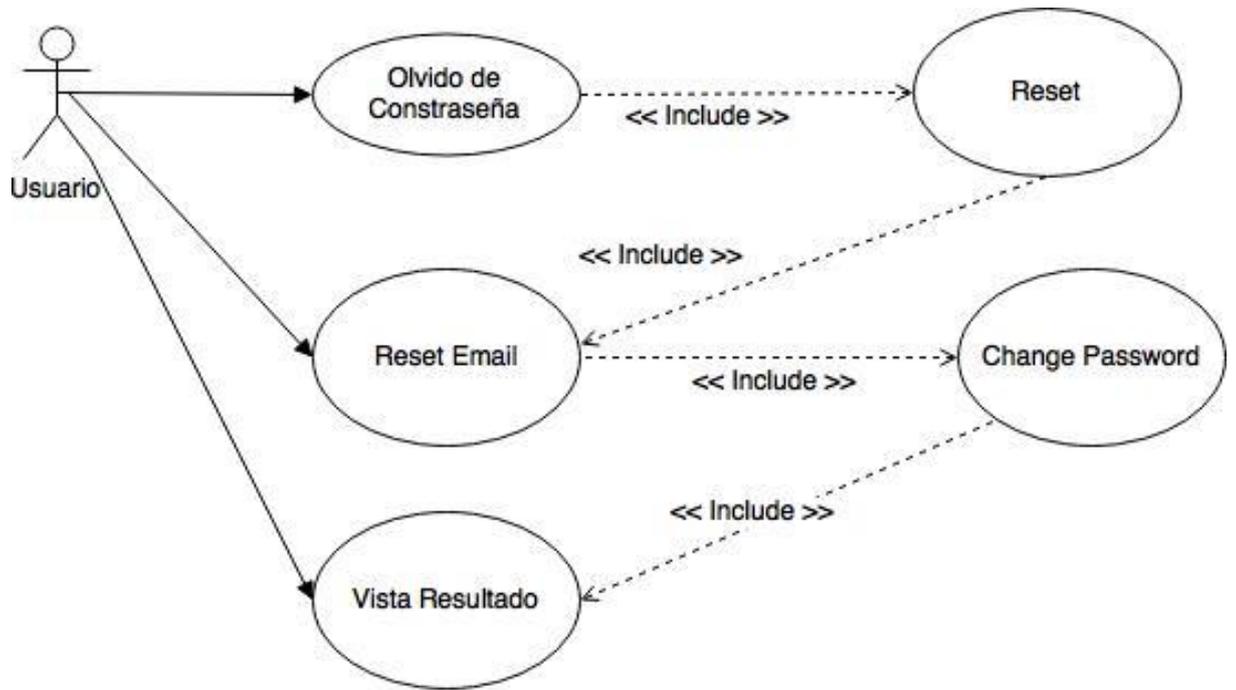


El usuario selecciona en el explorador de archivos que imagen quiere que sea procesada.



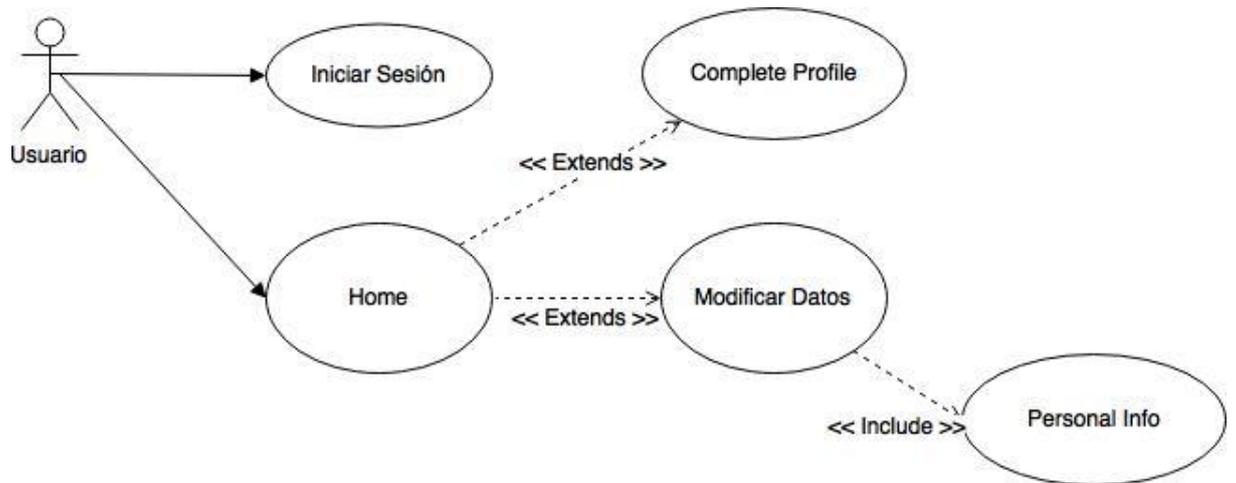
3. Cambiar contraseña:

El usuario puede cambiar su contraseña si el usuario olvida la actual.



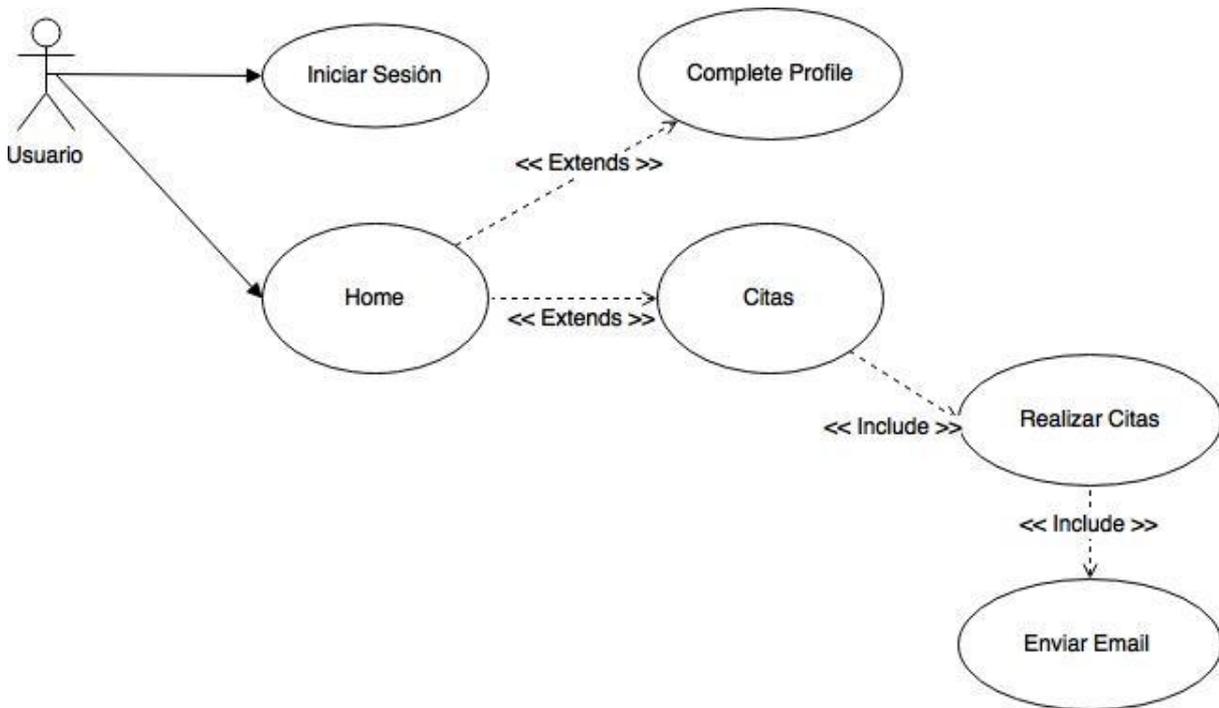
4. Modificar datos personales:

El usuario puede modificar sus datos personales una vez que ya haya completado su perfil en la pantalla de home podrá entrar a la opción de modificar sus datos.



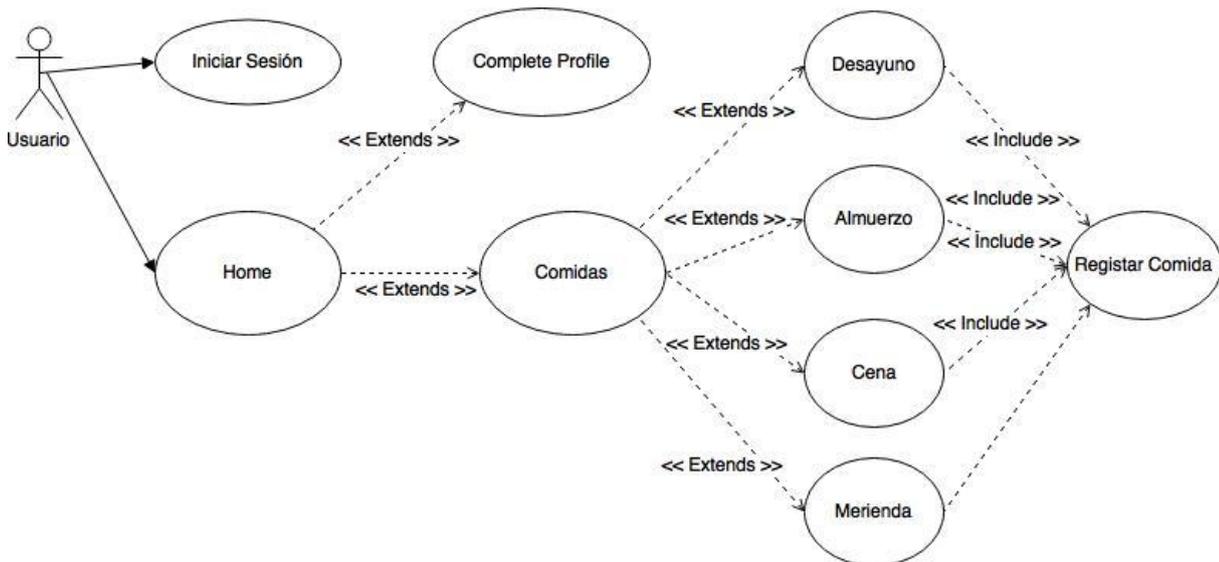
5. Concertar un cita:

El paciente puede concertar una cita con su doctor o el doctor puede planificar una cita con su paciente



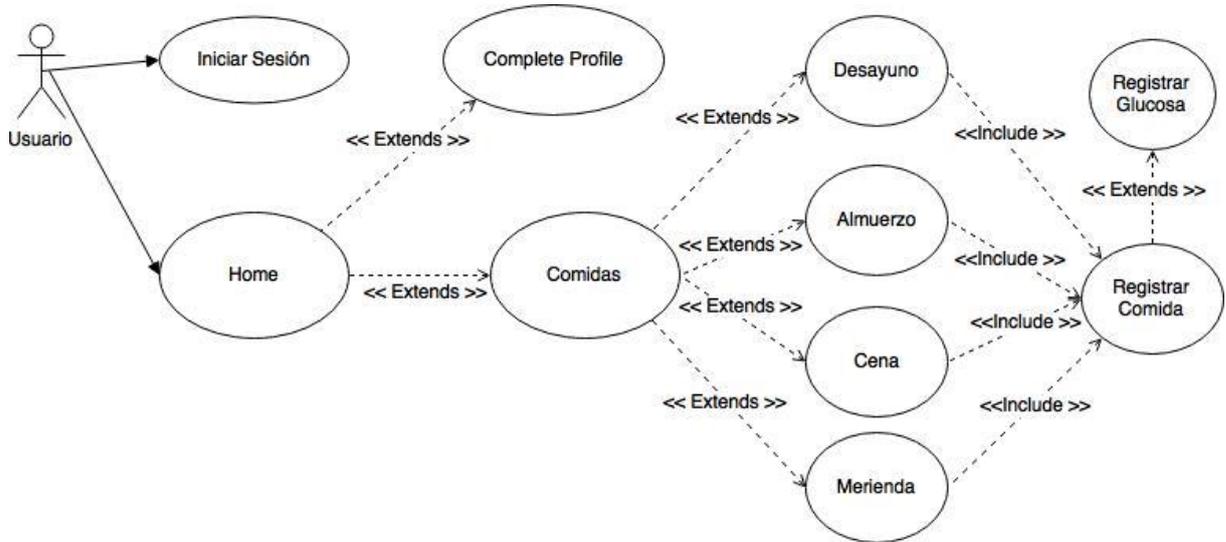
6. Registrar una Comida:

El paciente en el menú home tiene la opción de registrar una comida una vez seleccionada esa opción el paciente deberá elegir qué tipo de comida va a elegir, para luego registrar la comida.



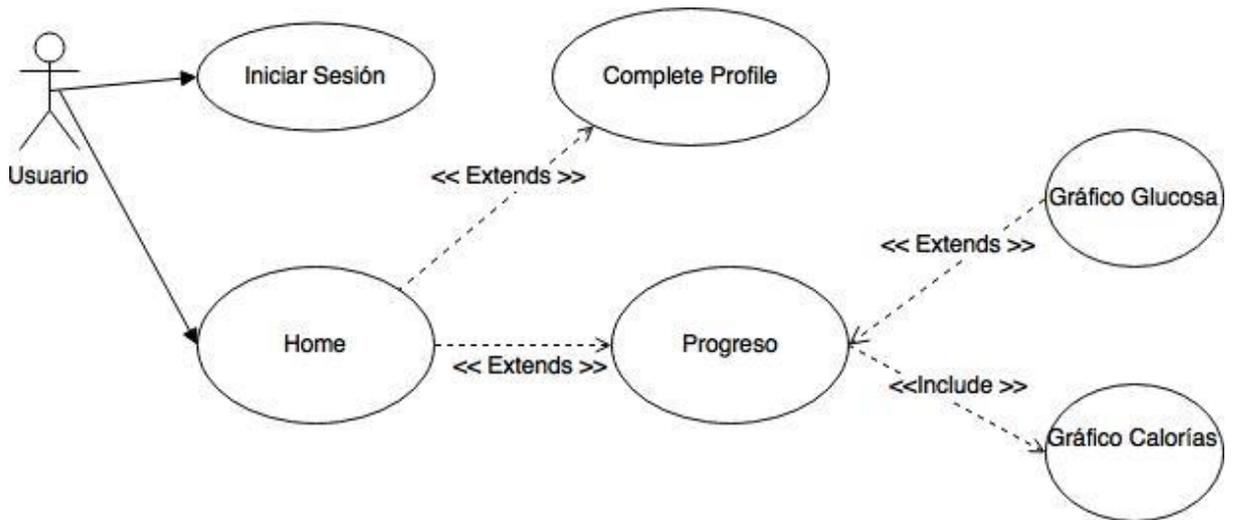
7. Registrar el nivel de glucosa:

El paciente solo si es diabético se registrará su nivel de glucosa antes de guardar su comida



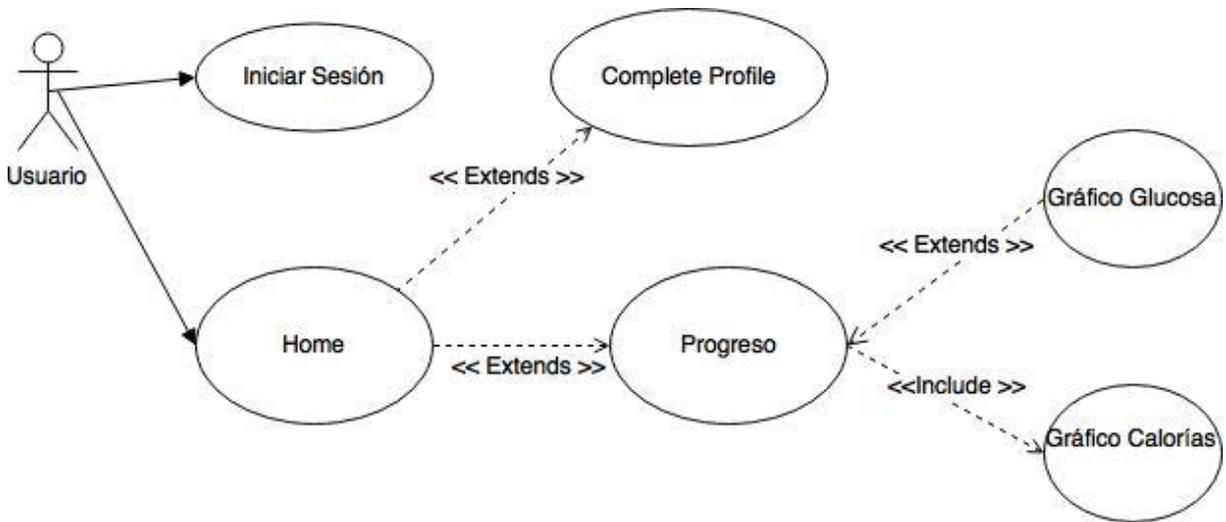
8. Consultar su ingesta calórica:

El paciente en el menú home tiene la opción de consultar su progreso alimenticio consultando por semana sus gráficos correspondientes



9. Consultar sus niveles de glucosa:

El paciente en el menú home tiene la opción de consultar su progreso alimenticio consultando por semana sus gráficos correspondientes el grafico de niveles de glucosa solo es desplegado si existen valores para la semana.



3.3 ARQUITECTURA DEL SISTEMA.

La arquitectura de Django es la anteriormente mencionada MTV.

Para empezar a entender MTV debemos fijarnos en la analogía con MVC.

- El *modelo* en Django sigue siendo **modelo**
- La *vista* en Django se llama **Plantilla (Template)**
- El *controlador* en Django se llama **Vista**

Los archivos predeterminados

Otra parte importante es entender el propósito de los archivos que se crean de manera predeterminada, estos son:

Archivos del proyecto

- **__init__.py**: Este es un archivo vacío que le dice a Python que debe considerar este directorio como un paquete de Python.
- **manage.py**: Este archivo contiene una porción de código que permite interactuar con el proyecto de Django de muchas formas.
- **settings.py**: Este archivo contiene todas las configuraciones para el proyecto, la conexión a nuestra base de datos, configuración de la zona horaria, correo electrónico, ubicación de nuestros archivos estáticos e imágenes, configuración para poder autenticarnos con las redes sociales, instalación de aplicaciones extra, etc.
- **urls.py**: Contiene las rutas que están disponibles en el proyecto, manejado por URLConf.

Archivos de la aplicación

- **__init__.py**: Este es un archivo vacío que le dice a Python que debe considerar este directorio como un paquete de Python.
- **models.py**: En este archivo se declaran las clases del modelo.
- **views.py**: En este archivo se declaran las funciones de la vista.
- **test.py**: En este archivo se declaran las pruebas necesarias para la aplicación, para mayor detalle sobre las pruebas y Django se puede consultar la documentación del proyecto.

La estructura de la aplicación es la siguiente:

```
food_control/
├── alimentus
│   ├── __init__.py
│   ├── __init__.pyc
│   ├── admin.py
│   ├── admin.pyc
│   ├── apps.py
│   ├── choices.py
│   ├── choices.pyc
│   ├── forms.py
│   ├── forms.pyc
│   ├── migrations
│   │   ├── 0001_initial.py
│   │   ├── 0001_initial.pyc
│   │   ├── 0002_auto_20170817_2308.py
│   │   ├── 0002_auto_20170817_2308.pyc
│   │   ├── 0003_auto_20170817_2335.py
│   │   ├── 0003_auto_20170817_2335.pyc
│   │   ├── __init__.py
│   │   └── __init__.pyc
│   ├── models.py
│   ├── models.pyc
│   ├── pipeline.py
│   ├── tests.py
│   ├── views.py
│   └── views.pyc
├── food_control
│   ├── __init__.py
│   ├── __init__.pyc
│   ├── settings.py
│   ├── settings.pyc
│   ├── urls.py
│   └── urls.pyc
```

```
| | | wsgi.py
| | | wsgi.pyc
| | | manage.py
| | | requirements.txt
| | | static
| | | | admin
| | | | | css
| | | | | | base.css
| | | | | | changelists.css
| | | | | | dashboard.css
| | | | | | fonts.css
| | | | | | forms.css
| | | | | | login.css
| | | | | | rtl.css
| | | | | | | widgets.css
| | | | | fonts
| | | | | | LICENSE.txt
| | | | | | README.txt
| | | | | | Roboto-Bold-webfont.woff
| | | | | | Roboto-Light-webfont.woff
| | | | | | | Roboto-Regular-webfont.woff
| | | | | img
| | | | | | LICENSE
| | | | | | README.txt
| | | | | | calendar-icons.svg
| | | | | | gis
| | | | | | | move_vertex_off.svg
| | | | | | | | move_vertex_on.svg
| | | | | | | icon-addlink.svg
| | | | | | | icon-alert.svg
| | | | | | | icon-calendar.svg
| | | | | | | icon-changelink.svg
| | | | | | | icon-clock.svg
| | | | | | | icon-deletelink.svg
```

- | | | | └─ icon-no.svg
- | | | | └─ icon-unknown-alt.svg
- | | | | └─ icon-unknown.svg
- | | | | └─ icon-yes.svg
- | | | | └─ inline-delete.svg
- | | | | └─ search.svg
- | | | | └─ selector-icons.svg
- | | | | └─ sorting-icons.svg
- | | | | └─ tootag-add.svg
- | | | | └─ tootag-arrowright.svg
- | | | └─ js
- | | | | └─ SelectBox.js
- | | | | └─ SelectFilter2.js
- | | | | └─ actions.js
- | | | | └─ actions.min.js
- | | | | └─ admin
- | | | | | └─ DateTimeShortcuts.js
- | | | | | └─ RelatedObjectLookups.js
- | | | | └─ calendar.js
- | | | | └─ collapse.js
- | | | | └─ collapse.min.js
- | | | | └─ core.js
- | | | | └─ inlines.js
- | | | | └─ inlines.min.js
- | | | | └─ jquery.init.js
- | | | | └─ prepopulate.js
- | | | | └─ prepopulate.min.js
- | | | | └─ timeparse.js
- | | | | └─ urlify.js
- | | | | └─ vendor
- | | | | | └─ jquery
- | | | | | | └─ LICENSE-JQUERY.txt
- | | | | | | └─ jquery.js
- | | | | | | └─ jquery.min.js

- | | | | └─ xregexp
- | | | | | └─ LICENSE-XREGEXP.txt
- | | | | | └─ xregexp.min.js
- | | └─ css
- | | | └─ bootstrap-datetimepicker.css
- | | | └─ bootstrap-social.css
- | | | └─ bootstrap-theme.css
- | | | └─ bootstrap-theme.css.map
- | | | └─ bootstrap-theme.min.css
- | | | └─ bootstrap-theme.min.css.map
- | | | └─ bootstrap.css
- | | | └─ bootstrap.css.map
- | | | └─ bootstrap.min.css
- | | | └─ bootstrap.min.css.map
- | | | └─ datetimepicker.css
- | | | └─ dropdowns-enhancement.css
- | | | └─ dropdowns-enhancement.css.txt
- | | | └─ font-awesome-brands.css
- | | | └─ font-awesome-core.css
- | | | └─ font-awesome-light.css
- | | | └─ font-awesome-regular.css
- | | | └─ font-awesome-solid.css
- | | | └─ font-awesome.css
- | | | └─ font-awesome.min.css
- | | | └─ ie10-viewport-bug-workaround.css
- | | | └─ navbar-fixed-top.css
- | | | └─ optionalStyling.css
- | | | └─ signin.css
- | | | └─ web2cal.css
- | | | └─ webcalendar
- | | | | └─ font-awesome-4.7.0
- | | | | | └─ HELP-US-OUT.txt
- | | | | | └─ css
- | | | | | └─ font-awesome.min.css

- | | | | └─ fonts
- | | | | | └─ FontAwesome.otf
- | | | | | └─ fontawesome-webfont.eot
- | | | | | └─ fontawesome-webfont.svg
- | | | | | └─ fontawesome-webfont.ttf
- | | | | | └─ fontawesome-webfont.woff
- | | | | | └─ fontawesome-webfont.woff2
- | | | | └─ less
- | | | | | └─ animated.less
- | | | | | └─ bordered-pulled.less
- | | | | | └─ core.less
- | | | | | └─ fixed-width.less
- | | | | | └─ font-awesome.less
- | | | | | └─ icons.less
- | | | | | └─ larger.less
- | | | | | └─ list.less
- | | | | | └─ mixins.less
- | | | | | └─ path.less
- | | | | | └─ rotated-flipped.less
- | | | | | └─ screen-reader.less
- | | | | | └─ stacked.less
- | | | | | └─ variables.less
- | | | | └─ scss
- | | | | | └─ _animated.scss
- | | | | | └─ _bordered-pulled.scss
- | | | | | └─ _core.scss
- | | | | | └─ _fixed-width.scss
- | | | | | └─ _icons.scss
- | | | | | └─ _larger.scss
- | | | | | └─ _list.scss
- | | | | | └─ _mixins.scss
- | | | | | └─ _path.scss
- | | | | | └─ _rotated-flipped.scss
- | | | | | └─ _screen-reader.scss

- | | | | — _mixins.scss
- | | | | — _path.scss
- | | | | — _rotated-flipped.scss
- | | | | — _screen-reader.scss
- | | | | — _stacked.scss
- | | | | — _variables.scss
- | | | | — font-awesome.scss
- | | — fonts
- | | | | — 7e9aedd97b5ec4590edb8281ff12b168_favicon.ico
- | | | | — FontAwesome.otf
- | | | | — HelveticaNeueLTStd-Cn.woff.otf
- | | | | — Trade.woff
- | | | | — fontawesome-brands-900.eot
- | | | | — fontawesome-brands-900.otf
- | | | | — fontawesome-brands-900.svg
- | | | | — fontawesome-brands-900.ttf
- | | | | — fontawesome-brands-900.woff
- | | | | — fontawesome-brands-900.woff2
- | | | | — fontawesome-light-300.eot
- | | | | — fontawesome-light-300.otf
- | | | | — fontawesome-light-300.svg
- | | | | — fontawesome-light-300.ttf
- | | | | — fontawesome-light-300.woff
- | | | | — fontawesome-light-300.woff2
- | | | | — fontawesome-regular-400.eot
- | | | | — fontawesome-regular-400.otf
- | | | | — fontawesome-regular-400.svg
- | | | | — fontawesome-regular-400.ttf
- | | | | — fontawesome-regular-400.woff
- | | | | — fontawesome-regular-400.woff2
- | | | | — fontawesome-solid-900.eot
- | | | | — fontawesome-solid-900.otf
- | | | | — fontawesome-solid-900.svg
- | | | | — fontawesome-solid-900.ttf

- | | | — fontawesome-solid-900.woff
- | | | — fontawesome-solid-900.woff2
- | | | — fontawesome-webfont.eot
- | | | — fontawesome-webfont.svg
- | | | — fontawesome-webfont.ttf
- | | | — fontawesome-webfont.woff
- | | | — fontawesome-webfont.woff2
- | | | — glyphicons-halflings-regular.eot
- | | | — glyphicons-halflings-regular.svg
- | | | — glyphicons-halflings-regular.ttf
- | | | — glyphicons-halflings-regular.woff
- | | | — glyphicons-halflings-regular.woff2
- | | | — logo.png
- | | | — logo2.jpg
- | | | — logo3.png
- | | | — test.woff
- | | — images
- | | | — bars-chart.svg
- | | | — beans.svg
- | | | — breakfast.svg
- | | | — butcher.svg
- | | | — calendar-4.svg
- | | | — cereals.svg
- | | | — chips.svg
- | | | — chocolate.svg
- | | | — contact.svg
- | | | — contacts.svg
- | | | — cutlery.svg
- | | | — dish.svg
- | | | — egg.svg
- | | | — fruit-salad.svg
- | | | — id-card-2.svg
- | | | — juice.svg
- | | | — milk-3.png

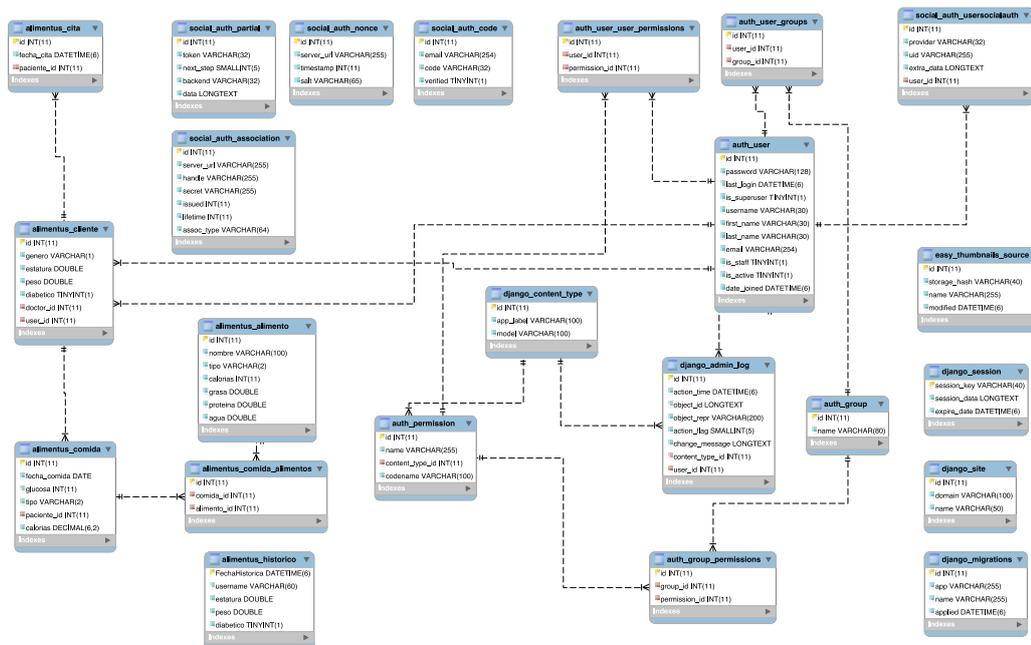
- | | | — milk-3.svg
- | | | — milk.png
- | | | — milk.svg
- | | | — olive-oil.svg
- | | | — prawn.svg
- | | | — salad.svg
- | | | — salt-and-pepper.svg
- | | | — spaghetti.svg
- | | | — steak-2.svg
- | | | — steak.svg
- | | | — water-bottle.svg
- | | — js
- | | | — DateTimeShortcuts.js
- | | | — RelatedObjectLookups.js
- | | | — SelectBox.js
- | | | — SelectFilter2.js
- | | | — actions.js
- | | | — actions.min.js
- | | | — bootstrap-datetimepicker.js
- | | | — bootstrap-datetimepicker.min.js
- | | | — bootstrap.min.js
- | | | — calendar.js
- | | | — collapse.js
- | | | — collapse.min.js
- | | | — core.js
- | | | — d3.js
- | | | — d3.min.js
- | | | — dropdowns-enhancement.js
- | | | — fontawesome.js
- | | | — ie-emulation-modes-warning.js
- | | | — ie10-viewport-bug-workaround.js
- | | | — inlines.js
- | | | — inlines.min.js
- | | | — jquery.init.js

- | |— jquery.js
- | |— jquery.min.js
- | |— jquery.min.map
- | |— moment-with-locales.js
- | |— moment.js
- | |— packs
 - | |— brands.js
 - | |— light.js
 - | |— regular.js
 - | |— solid.js
- | |— prepopulate.js
- | |— prepopulate.min.js
- | |— test.js
- | |— timeparse.js
- | |— transition.js
- | |— urlify.js
- | |— v4-shim.js
- | |— v4-shims.js
- | |— webcalendar
 - | |— es.js
 - | |— fullcalendar.js
 - | |— jquery-ui.min.js
 - | |— locale-all.js
- |— templates
 - |— activate.html
 - |— activate_account.html
 - |— activation_complete.html
 - |— alimento.html
 - |— base.html
 - |— charts.html
 - |— citas.html
 - |— consulta_pacientes.html
 - |— home.html
 - |— informacion.html

- └─ login.html
- └─ logout.html
- └─ messages.html
- └─ navbar.html
- └─ password_reset_confirm.html
- └─ password_reset_email.html
- └─ password_reset_form.html
- └─ recordatorio_cita.html
- └─ registrar_comida.html
- └─ sidebar.html
- └─ signup.html
- └─ tipo_comida.html

32 carpetas, 318 archivos componen nuestra aplicación

A continuación, se explicará nuestro modelo de entidad relación que ha sido creado a través de la funcionalidad ORM de Django.



Django al hacer la primera migración, crea en nuestra base de datos todas las entidades responsables del manejo de sesiones, usuarios, permisos, dominios, migraciones, log del sistema, etc.

En la aplicación se ha instalado el paquete Python Social Auth este paquete nos brinda la funcionalidad de autenticarnos y crear un usuario a través del protocolo OAuth de las redes sociales, este paquete es casi plug and play para instalarlos simplemente hay que seguir los siguientes pasos

- Instalar el paquete en nuestro ambiente virtual de nuestra aplicación
- Realizar una migración inicial a nuestra base de datos
- Configurar los pipelines del paquete por defecto el paquete crea usuarios y es capaz de realizar el inicio de sesión por sí mismo, pero es posible modificar esto para lograr otros comportamientos
- Configuración de las redes sociales que vamos a usar, esto implica crear para cada red social una aplicación para así tener appID y AppSecret necesarios para la autenticación OAuth

Python Social Auth crea en nuestra base de datos 5 tablas: social_auth_association, social_auth_code, social_auth_code, social_auth_nonce, social_auth_partial, social_auth_usersocial. De estas 5 tablas las utilizadas serían social_auth_partial y social_auth_usersocial.

social_auth_partial: se almacena el token generado por la red social, la red social y la respuesta del servidor donde contiene username, email, nombre y otros datos que no son utilizados en nuestra aplicación

social_auth_usersocial: se almacena el username con el que sea ha creado el usuario, red social a la que pertenece y se almacena el foreign key de la tabla auth_user así están ligados los usuarios Django con los usuarios creados a través de Python Social Auth.

Las tablas utilizadas para la aplicación :

- Cliente: esta tabla está ligada al auth_user uno a uno, almacenamos, genero, estatura, peso, si la persona padece diabetes , doctor al cual está asignado.

- Cita: en esta tabla almacenaremos el paciente que ha solicitado una cita y el id del paciente.
- Alimento: en esta tabla almacenamos los alimentos que posee nuestra base de datos: nombre, calorías (por cada 100 gramos), grasa, proteína, agua y tipo de alimento que están clasificados de la siguiente forma:
 - Lácteos
 - Huevos
 - Cárnicos
 - Pescados, moluscos, reptiles, crustáceos
 - Grasas y Aceites
 - Cereales
 - Legumbres, semillas y frutos secos
 - Verduras y Hortalizas
 - Frutas Azúcar y Chocolates
 - Bebidas
 - Producto de uso nutricional
- Comida: en esta tabla se almacenará las comidas realizadas almacenaremos los alimentos consumidos, la cantidad de calorías almacenadas en esa comida, fecha en la cual se realizó, el paciente que la realizó, nivel de glucosa y tipo de comida: desayuno, almuerzo, cena y merienda.

3.4 IMPLEMENTACIÓN Y FUNCIONAMIENTO DEL SISTEMA.

3.4.1 IMPLEMENTACIÓN.

Como hemos mencionado anteriormente en el archivo `views.py` es donde se realiza la funcionalidad de las vistas, aquí se hacen las llamadas a la base de datos, se manipulan los datos y se crean las vistas utilizando las páginas html almacenadas en la carpeta `template`. A continuación, explicaremos en detalle que hace cada método en el fichero `views.py`.

El primer método que está en el fichero es **login**. Este método funciona de dos formas distintas si es llamado por un doctor y la autenticación es correcta se redirige a la vista principal del doctor que es todos los pacientes que posee con sus datos correspondientes.

Si el método es invocado por un paciente si el resultado es correcto verificamos que el paciente tenga los datos completos si no los tiene a la vista `home` se envía una variable de control para que en la vista `home` se pida que rellene los datos del paciente, si el perfil está completo se reenvía a la vista `home`. Si la autenticación no es correcta se reenvía a la vista `login` indicando con un mensaje por pantalla que el usuario o contraseña son inválidos, si el usuario es válido pero no se ha activado se reenviara a la vista de `login` indicando que contacte con el administrador, deberá reenviarse un correo para que el usuario pueda activar su cuenta.

Signup: este método es para la creación de una cuenta, en este método se insertará en la tabla `auth_user` y en la tabla `cliente`, si los datos están correctos se almacenará en la base de datos y se enviara un correo al paciente para que active su cuenta, este correo utiliza seguridad `uid64` para encriptar la clave primaria del usuario y con el objeto `usuario` se genera un token, estos dos valores se agregan en la url que hay que pinchar para activar el usuario. Si los datos son incorrectos se reenviará a la pantalla de `signup` de nuevo.

Activate: este método es responsable de activar tu cuenta en el sistema para así poder iniciar sesión en la aplicación. Recibe por parámetro un objeto encriptado con `uid64` y un token, se, desencripta el objeto `uid64` que contiene la clave primaria, se hace una consulta que nos devolverá el usuario, luego hacemos un check con el usuario que tenemos ahora en memoria y con el token que hemos recibido si son iguales todo está en orden y se activa la cuenta del usuario, lo redirigimos a la vista `home` y mostramos un mensaje de bienvenida.

Reset: este método es responsable de enviar un correo electrónico al usuario que ha olvidado su contraseña. Primero se buscará si el nombre del usuario existe de ser así se envía un correo electrónico indicando las instrucciones que hay que seguir para cambiar la contraseña, este correo electrónico tiene un link en el que hay que pinchar, tiene dos variables que están encriptadas con uid64 una es la clave primaria del usuario y la otra el timestamp del momento en el que ha sido enviado el correo. Si el usuario no existe se reenvía a la vista de login indicando un error, si el usuario se ha dado de alta a través de una red social esta opción no está disponible.

Change_password: este método es responsable de cambiar el password de la persona una vez que haya utilizado el link que le ha sido enviado por correo electrónico una vez que se cambie el password este nuevo password se cambiara para acceder a la aplicación.

Reset_confirm: este método es responsable de validar el link que se envía a través de correo electrónico y reenviar a la vista responsable de cambiar el password. Recibe por parámetro dos variables encriptadas, la primera contiene la clave primaria del usuario, con ella se busca el usuario y luego se valida que no haya pasado más de 15 minutos del envío del correo a ser utilizado, esto es por razones de seguridad si estas dos validaciones son superadas se redirige a la vista de cambio de contraseña, de lo contrario se notifica que el link ya no es valido

Validate_username; este método es responsable de buscar si el nombre de usuario ya existe en la base de datos y devuelve en json una respuesta si el usuario existe, este método es utilizado en el template signup para verificar la existencia del usuario.

Home: este método es la vista de inicio de la aplicación tiene dos funcionalidades como paciente y como doctor.

Si se accede como doctor a la aplicación su vista de inicio es una tabla que indica la cantidad de pacientes asignado a él y luego cada fila muestra el username, nombre, apellido, email, peso, estatura, si es diabético y cuantas citas tiene concertadas.

Si se accede como paciente a la aplicación primero se valida que su perfil este completo de no serlo se envía una variable de control que obliga al usuario a completar su perfil.

Log_out: este método es responsable de terminar la sesión de un usuario.

Complete_profile: este método se encarga de completar el perfil de un paciente, esto ocurre cuando el paciente se da de alta a través de la aplicación y no por red social.

Citas: este método es responsable de realizar las citas en la aplicación los datos son recibidos como dos listas de eventos guardados tipo json, siguiendo la estructura de los eventos de Full Calendar nombrado anteriormente. Este método realiza varias validaciones, si el paciente ya tiene cita para ese día o si para ese día ya existen más de 8 citas para ese día tampoco guardara las citas para ese día, de pasar estas dos validaciones se guardarán las citas. Este método también es responsable de borrar las citas haya sido elegidas para ello.

Signup_doctor: este método es responsable de dar de alta a un paciente, a través de la aplicación siendo doctor.

Crear_alimento: este método es responsable de dar de alta un alimento, solo disponible si eres doctor.

Personal_info: este método es responsable de modificar los datos personales de un paciente.

Get_user: este método es responsable de devolver un paciente a través de su username.

Get_doc: este método es responsable de devolver un doctor a través de su username.

Get_food: este método es responsable de devolver un alimento a través de su clave primaria

Busca_paciente: este método es responsable de devolver todos los pacientes de un doctor.

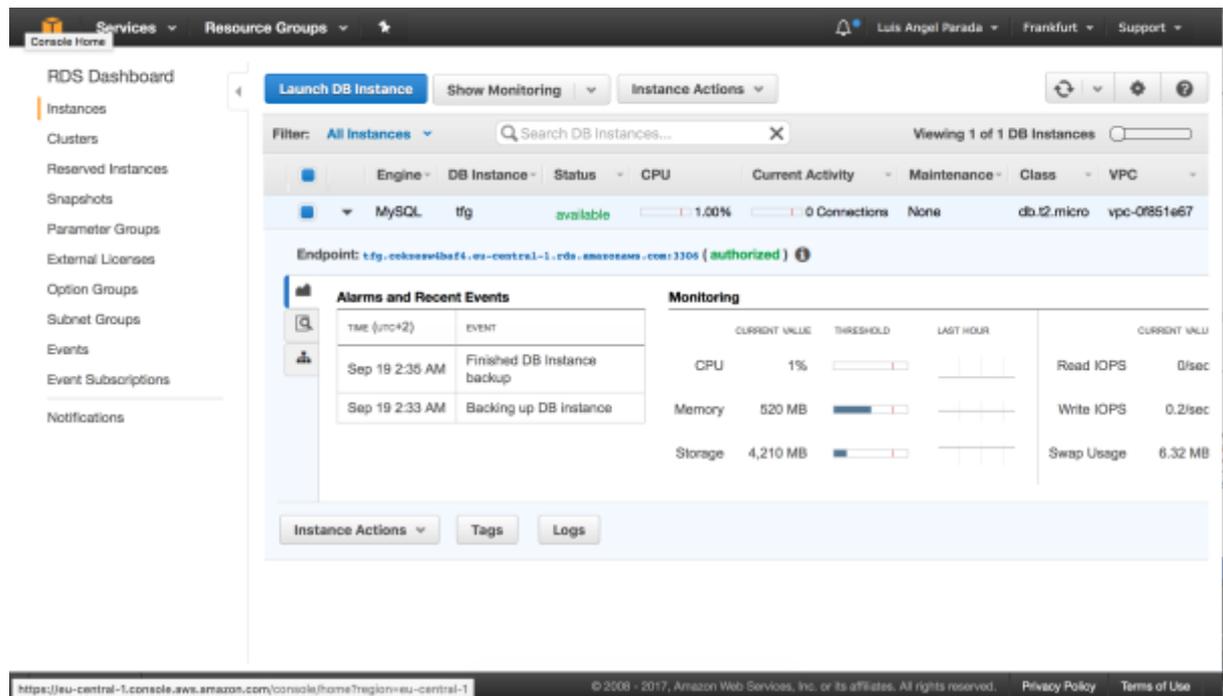
Registrar_comida: este método es responsable de guardar una comida en el sistema, recibe como parámetro el tipo de comida, los alimentos los recibe en una lista de tipo json que cada objeto contiene la clave primaria del alimento y la cantidadde gramos de cada alimento, se procesa cada alimento para conocer el valor de las calorías ingeridas.

nivelesQuery_alimentos: este método es responsable de devolver una lista json de alimentos filtrados por tipo de alimento.

Progress_charts: este método es responsable de generar los datos necesarios para crear los gráficos correspondientes de calorías o niveles de glucosa por semana, este método recibe el username.

3.4.2 DESPLIEGUE A PRODUCCIÓN

Para la subida a producción de la aplicación en Amazon Web Services, lo primero fue crearnos una cuenta en la plataforma y elegir la región de Europa Central-1 que es la sede es Frankfurt por cuestiones de latencia y cercanía se ha elegido este lugar aunque el servicio se ofrece en gran variedad de lugares para el despliegue. Una vez que tenemos nuestra cuenta creada es necesario levantar primero la instancia de nuestra base de datos, que por nombre llevará el nombre de BEDCA_TFG y será una instancia de MySQL 5.7 que se ejecutará en el puerto 3306, al completar el proceso de creación de base de datos esperamos unos minutos para que la instancia se levante.



The screenshot shows the AWS RDS console interface. The left sidebar contains navigation options like 'Instances', 'Clusters', and 'Snapshots'. The main content area displays the details for an RDS instance named 'tfg'. The 'Details' tab is active, showing various configuration and security parameters.

Configuration Details		Security and Network	
ARN	arn:aws:rds:eu-central-1:aws:iam::3326:authorized	Availability Zone	eu-central-1a
Engine	MySQL 5.7.16	VPC	vpc-0f851e07
License Model	General Public License	Subnet Group	default (Complete)
Created Time	May 18, 2017 at 9:24:24 PM UTC+2	Subnets	subnet-680a2722 subnet-0f7c5b07 subnet-1dc04a767
DB Name	BEDCA_TFG	Security Groups	rds-launch-wizard (sg-7bdb4b10) (active)
Username	root	Publicly Accessible	Yes
Option Group	default:mysql-5-7 (in-sync)	Endpoint	tfg-eu-central-1.rds.amazonaws.com
Parameter Group	default:mysql5.7 (in-sync)	Port	3306
Copy Tags To Snapshots	No	Certificate Authority	rds-ca-2015 (Mar 5, 2020)
Resource ID	db-G2YYJHMMNQFD43X3NMH5KE2C	Instance and IOPS	
IAM DB Authentication Enabled	No	Instance Class	db.t2.micro
		Storage Type	General Purpose (SSD)
		IOPS	disabled
		Storage	5 GB

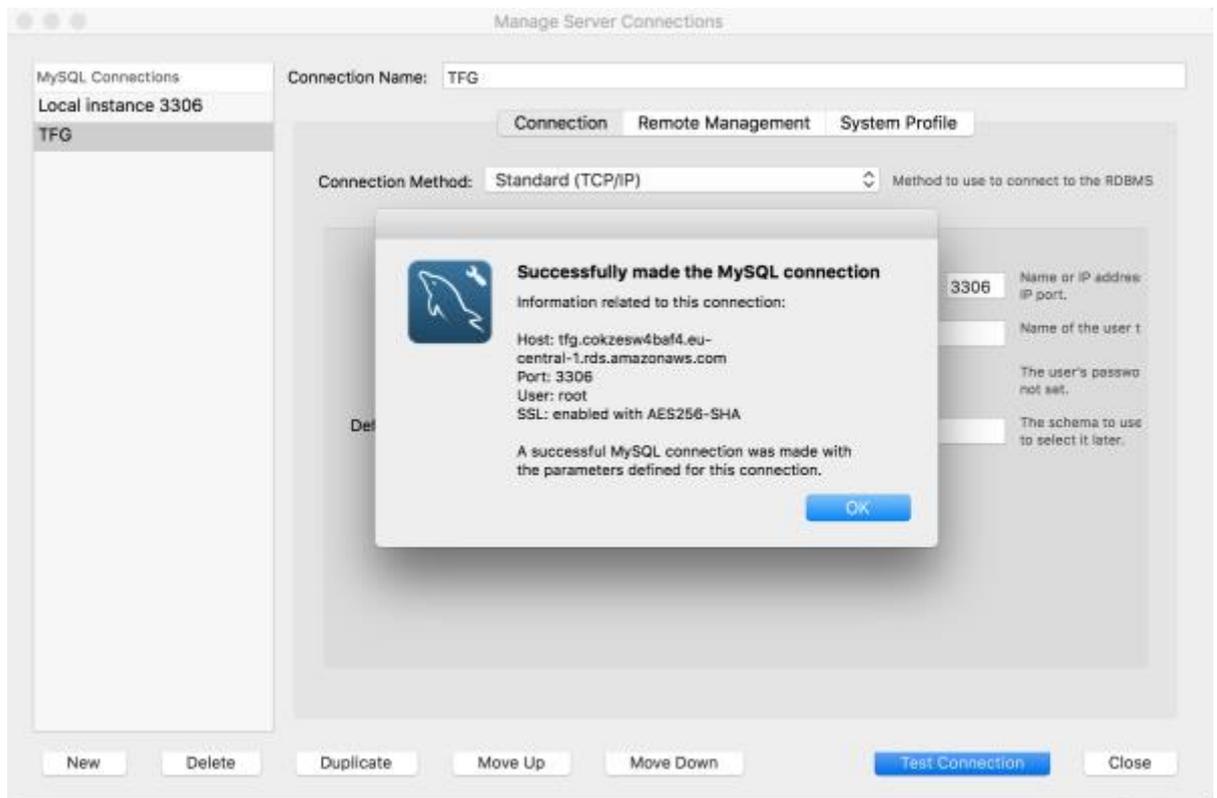
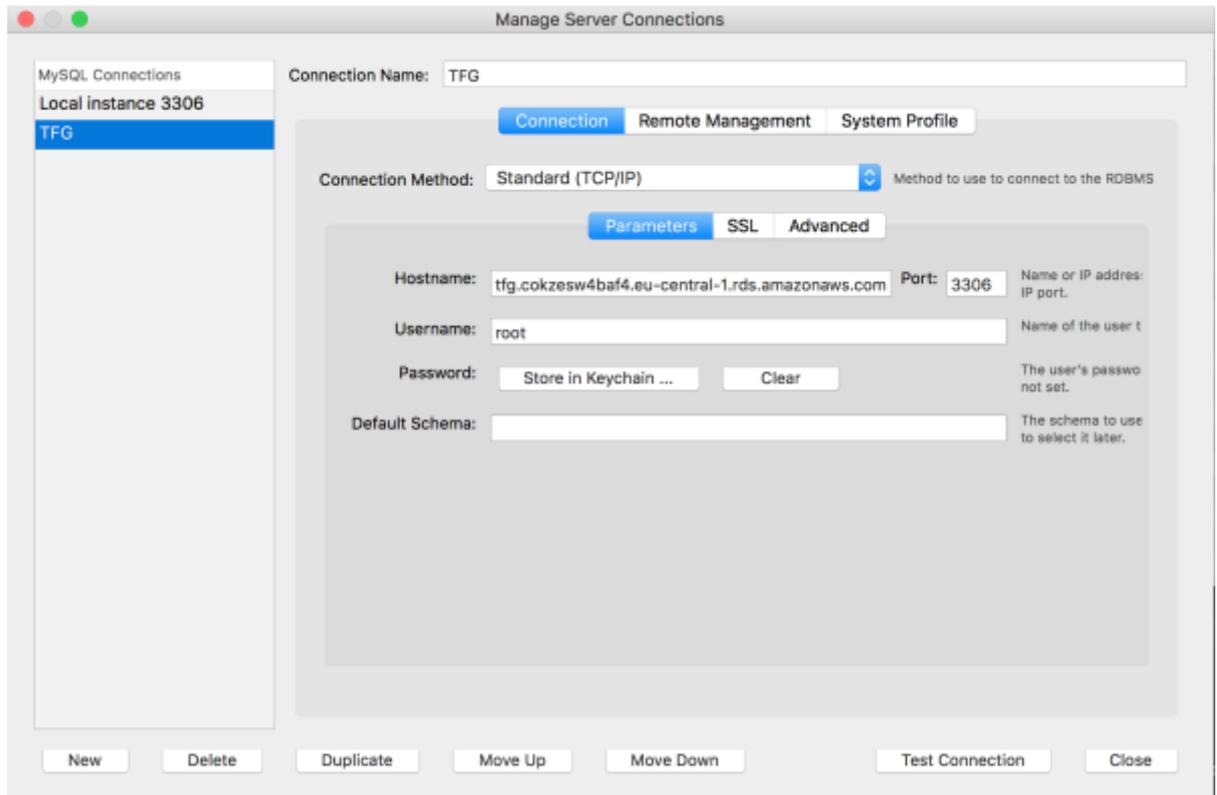
Monitoring Details: Enhanced Monitoring Enabled **No**

This screenshot shows the 'Tags' and 'Maintenance Details' sections of the RDS instance 'tfg'. The 'Tags' section includes an 'Add/Edit Tags' button and a table with one tag: 'workload-type' with the value 'other'. The 'Maintenance Details' section shows the instance's status and scheduled maintenance windows.

Encryption Details		Availability and Durability		Maintenance Details	
Encryption Enabled	No	DB Instance Status	available	Auto Minor Version Upgrade	Yes
		Multi AZ	No	Maintenance Window	sun-02:17-04:02:47
		Automated Backups	Enabled (1 Day)	Backup Window	00:22-00:52
		Latest Restore Time	September 19, 2017 at 6:10:00 PM UTC+2	Pending Maintenance	None

Key	Value	Remove
workload-type	other	✖

Luego de haber creado nuestra base de datos verificamos que funciona, configuramos una conexión a través de MySQLWorkbench y hacemos una prueba de conexión para verificar que está se conecta correctamente.



Podemos ver que funciona correctamente la base de datos, ahora proseguimos a crear nuestra instancia de Amazon Beanstalk que es nuestro contenedor para nuestra aplicación web, será una máquina virtual de Linux 64bits V2.5.3 que ejecuta una versión de Python 2.7 ya que nuestra aplicación Django fue desarrollada en esta versión.

Para preparar la aplicación para subirla es necesario instalar en nuestro ambiente virtual de Python que contiene nuestra aplicación el paquete awscli, una vez instalado nos autenticamos con nuestra cuenta de amazon web service.

Luego de haber instalado awscli hay que configurar la carpeta de la aplicación para la subida siguiendo estos pasos:

1. Situarse dentro de la aplicación a través de la consola y ejecutar la sentencia “pip freeze > requirements.txt” esto es un archivo que contiene todas las dependencias de nuestra aplicación, que luego son instaladas al subir nuestra aplicación.
2. Crear una carpeta oculta llamada “.ebextensions” que contendrá el archivo django.config donde se indica en que ruta esta nuestra aplicación.
3. Escribir desde la consola eb deploy y verificar que todo está correcto.

```
(BEDCA_TFG) luisangelparada@Luiss-MacBook-Pro:~/Documents/produccion/BEDCA_TFG/b
in/food_control$ eb deploy
Creating application version archive "app-170919_211532".
Uploading food_control3/app-170919_211532.zip to S3. This may take a while.
Upload Complete.
INFO: Environment update is starting.
INFO: Deploying new version to instance(s).
INFO: New application version was deployed to running EC2 instances.
INFO: Environment update completed successfully.
```

CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN

Conclusiones y futuras líneas de investigación.

CONCLUSIONES Y FUTURAS MEJORAS

Nuestra aplicación se centra en el desarrollo de una aplicación que sirva de apoyo para los nutricionistas y pacientes el llevar un control alimenticio de un paciente como también sus niveles de glucosa. Nuestra aplicación permite eso llevar un control alimenticio por semana de lo comido, así facilitando la tarea del nutricionista para hacer cambios de dietas a su paciente.

Mejora también el sistema de citas con el nutricionista ya que serviría como agenda electrónica para el doctor como para el paciente. Sirve de una plataforma de autoayuda para el paciente ya que siempre y cuando se mantenga la honestidad de sus comidas puede realizar consultas de lo que ha comido y cuantas calorías ha comido cada día de la semana, logrando así llevar una dieta balanceada

Este proyecto sirve de base para lograr una aplicación estándar para los nutricionistas en España que sea capaz de llevar un control alimenticio sobre sus pacientes. El futuro de esta aplicación sería la ampliación de la base de datos de alimenrasos mas precisa podrá ser el registro de estas comidas, el desarrollo de la aplicación en plataforma móvil Android y iOS, realizar algoritmos de capaces de detectar patrones alimenticios negativos que sean notificados al nutricionista para poder aplicar una corrección a tiempo, como también una recomendación automática sobre el mismo

PRESUPUESTO DEL PROYECTO

PRESUPUESTO DEL PROYECTO

El presupuesto para el presente proyecto se divide en el coste de materiales y el coste de la mano de obra.

COSTE DE LA MANO DE OBRA

CONCEPTO	HORAS	COSTE/HORA	COSTE TOTAL
INGENIERÍA			
Ingeniero Técnico en Informática (Jefe de Proyecto)	40	22	880
Analista Programador	80	12	960
Programador	550	10	5500
Total			7340

En el concepto de Ingeniería incluye:

Análisis de la aplicación en función de los requerimientos:

- Recogida de requisitos.
- Diseño y desarrollo de la aplicación.
- Desarrollo de la interface de usuario.
- Implementación de los Informes.

Depuración del código y puesta en marcha.
Redacción de la memoria.

COSTE DEL MATERIAL

El coste del hardware ya que utilizaremos Amazon Web Services es una plataforma de servicio en la nube que trabaja bajo demanda esto implica que no tendremos coste de hardware directo, solo el pago de las tarifas de los servicios contratados.

Este cálculo de presupuesto se hará asumiendo un tiempo de producción por año así podemos obtener un precio estimado de un año, si el negocio aumenta los servicios contratados también subirían su coste.

COSTE HARDWARE

DESCRIPCIÓN	NUMERO	COSTE/HORA	COSTE/DÍA	TOTAL
Amazon RDS para MySQL	1	0,03	0,72	262,8
Amazon Beanstalk con EC2	1	0,011	\$0,264	96,36
TOTAL				359,16

COSTE SOFTWARE

No existe coste de software ya que todos el software utilizado es de código abierto gratuito

DESCRIPCIÓN	TOTAL
COSTE HARDWARE	359,16
COSTE SOFTWARE	0
COSTE TOTAL DEL MATERIAL	359,16

GASTOS GENERALES

En este apartado se incluye diversos gastos:

Material de oficina (folios, bolígrafos, recambios de impresora, etc).

Desplazamientos, dietas, Comunicaciones, etc.

GASTOS GENERALES	300
------------------	-----

COSTE GLOBAL

COSTE MANO DE OBRA	7340
COSTE MATERIAL	359,16
GASTOS GENERALES	300
TOTAL	7999,16

EL Coste global del proyecto asciende a la cantidad de siete mil novecientos noventa y nueve con dieciseis.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

- [1] DjangoProject.com, Documentación oficial de Django:
<https://www.djangoproject.com/>
- [2] ThedjangoBook.com, Un resumen a fondo de Django:
<https://djangobook.com/tutorials/django-overview/>
- [3] ThedjangoBook.com, Un texto explicativo de porque Django sobre otros framework:
<https://djangobook.com/tutorials/why-django/>
- [4] Seedo.com, Prevención, diagnóstico y tratamiento de la obesidad. posicionamiento de la sociedad española para el estudio de la obesidad de 2016.
http://www.seedo.es/images/site/POSICIONAMIENTO_SEEDO2016_VC.pdf
- [5] aws.amazon.com, Documento explicativo sobre Amazon Web Services:
<https://d0.awsstatic.com/whitepapers/aws-overview.pdf>
- [6] Aws.amazon.com, Manual explicativo de cómo desplegar una aplicación Django Amazon Web Services.
<http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create-deploy-python-django.html>
- [7] Businessinsider.com, 10 aplicaciones móviles sobre salud:
<http://www.businessinsider.com/why-amazon-is-so-hard-to-topple-in-the-cloud-and-where-everybody-else-falls-2017-7>
- [8] Skyhighnetworks, Artículo sobre la cuota de mercado de la computación en la nube
<https://www.skyhighnetworks.com/cloud-security-blog/microsoft-azure-closes-iaas-adoption-gap-with-amazon-aws/>
- [9] Openinnova.es, Comparativa entre Amazon Web Services, Azure, Google Cloud.
<https://www.openinnova.es/amazon-aws-vs-microsoft-azure-vs-google-cloud-cual-elegir/>
- [10] Tiobe.com, Índice de los lenguajes de programación más utilizados en Septiembre 2017.
<https://www.tiobe.com/tiobe-index/>
- [11] Lifewire.com, Artículo sobre desarrollo web vs desarrollo móvil.
<https://www.lifewire.com/native-apps-vs-web-apps-2373133>
- [12] D3js.org, Página oficial de la librería D3.js
<https://d3js.org/>
- [13] bedca.net, Página oficial de Base de Datos Española de Composición de Alimentos
<http://www.bedca.net/>
- [14] redmonk.com, Índice de ranking de lenguajes de programación 2017
<http://redmonk.com/sograzy/2017/06/08/language-rankings-6-17/>

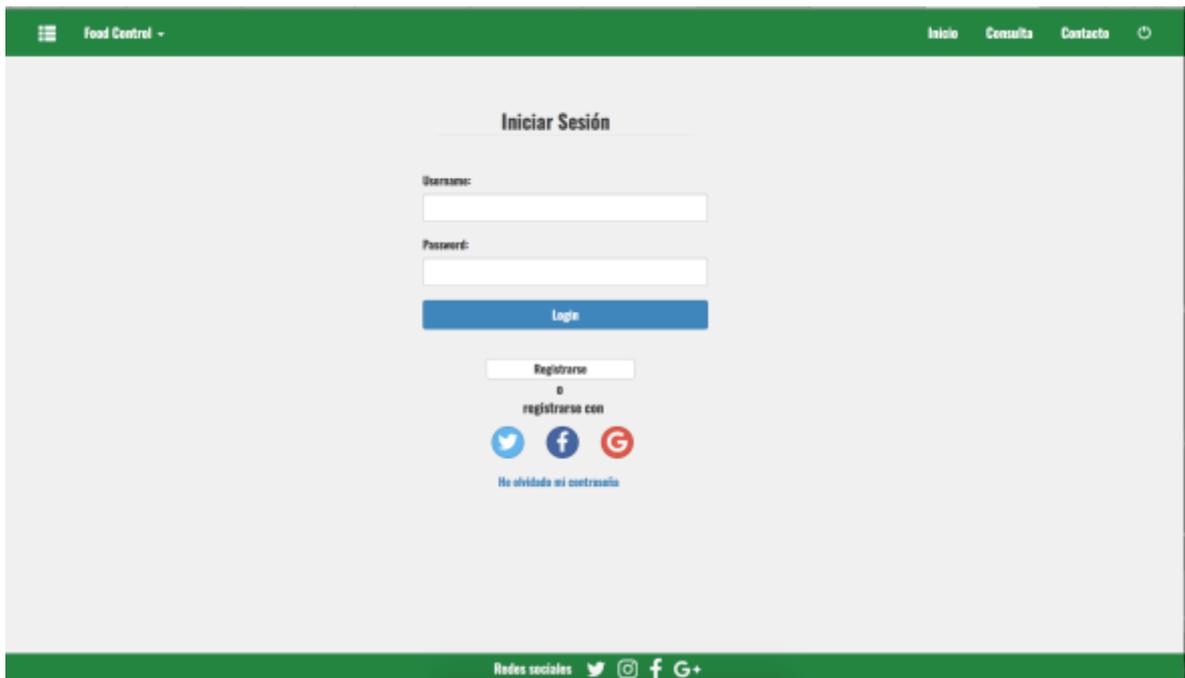
ANEXO: MANUAL DE USUARIO

MANUAL DE USUARIO

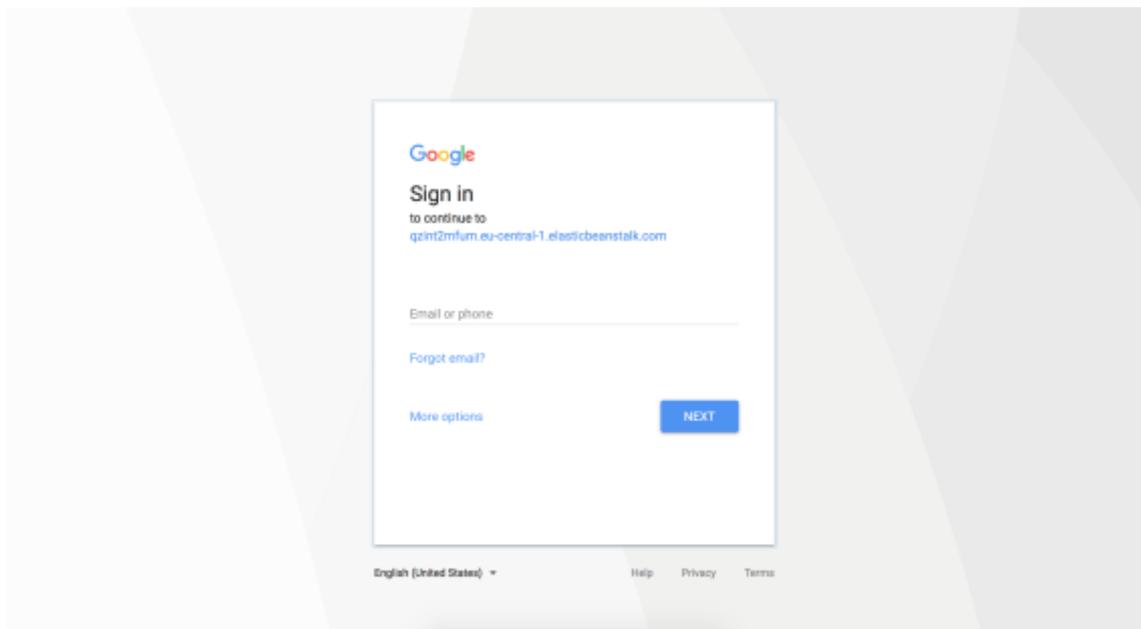
REGISTRO EN EL SISTEMA

Para registrarse en el sistema tenemos dos opciones a través de redes sociales o del formulario de la aplicación, a continuación, primero explicaremos como entrar por redes sociales.

Para crear una cuenta con redes sociales lo único que hay que realizar es pinchar nuestro icono que represente la red social de preferencia

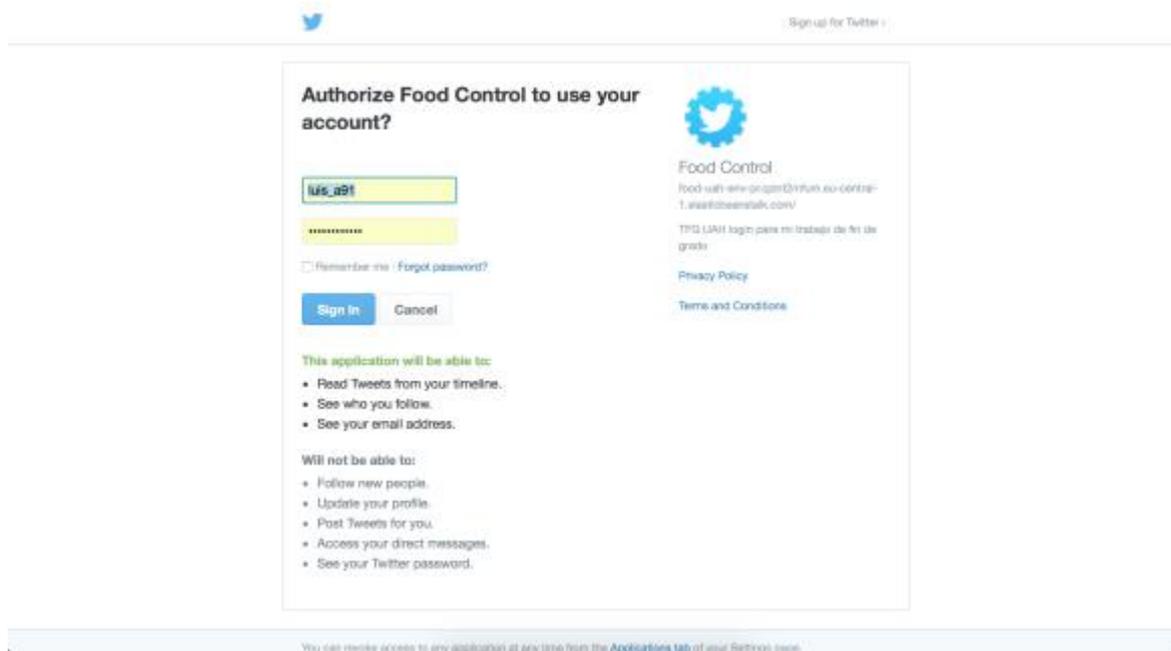


Google

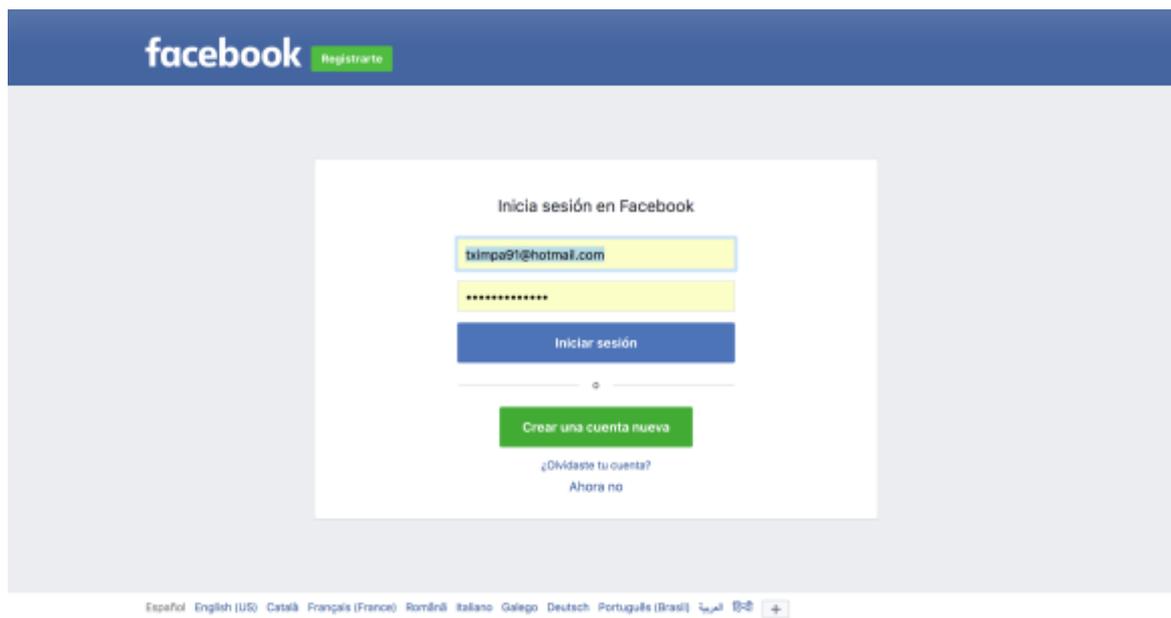


Manual de usuario.

Twitter



Facebook

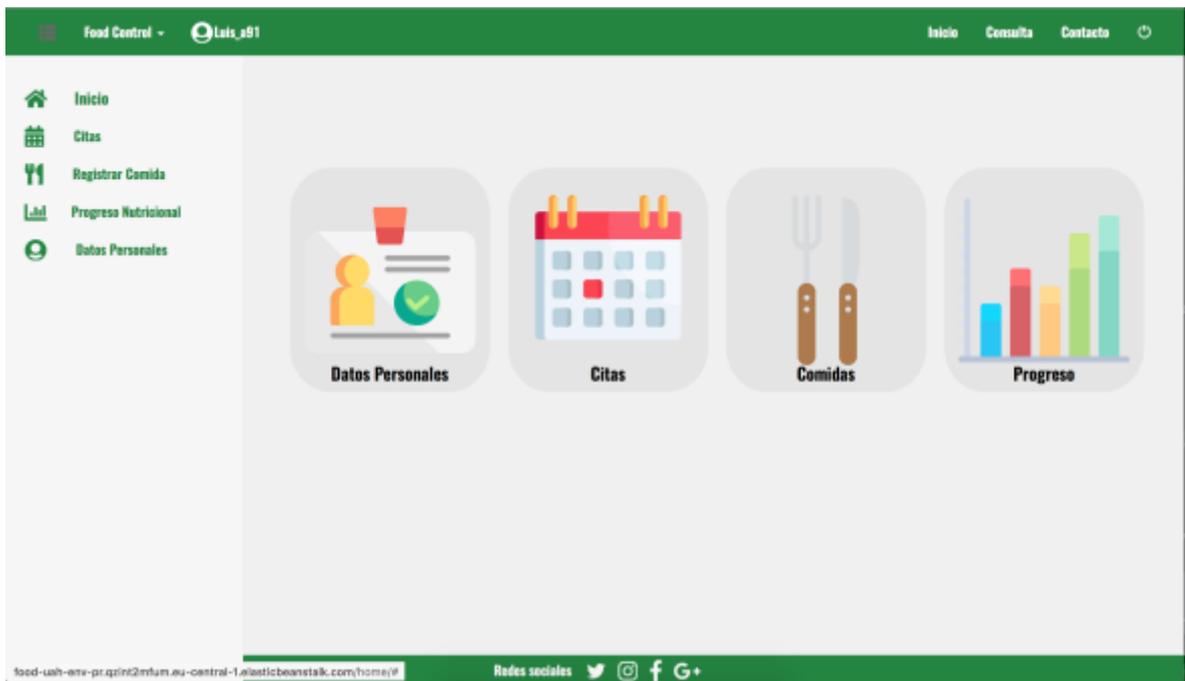


Una vez que nos autentiquemos con nuestra red social, se mostrará el menú principal de la aplicación y no pedirá que completemos nuestro perfil.

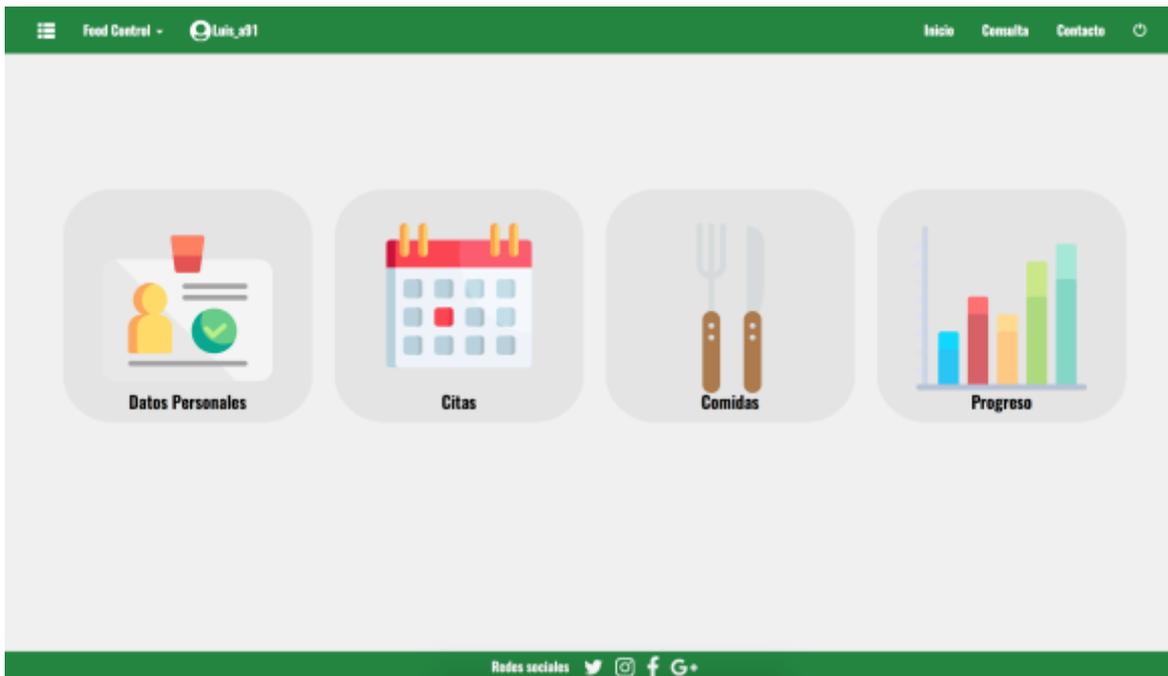
Manual de usuario.

The screenshot shows a web application interface with a green header. The header contains a menu icon, the text 'Food Control', a user profile icon with the name 'Gabrielazachado6', and navigation links for 'Inicio', 'Consulta', and 'Contacto'. A central modal window titled 'Completar perfil' is displayed. The modal has a subtitle 'Completa tu perfil completando los siguientes datos'. It contains four input fields: 'Doctor' (with a dropdown arrow), 'Genero' (with 'Masculino' selected and a dropdown arrow), 'Estatura' (with a dropdown arrow), and 'Peso' (with a dropdown arrow). Below these fields is a checkbox labeled 'Diabetico'. At the bottom of the modal is a blue button labeled 'Continuar'. The footer of the page includes the text 'Redes sociales' followed by icons for Twitter, Instagram, Facebook, and Google+.

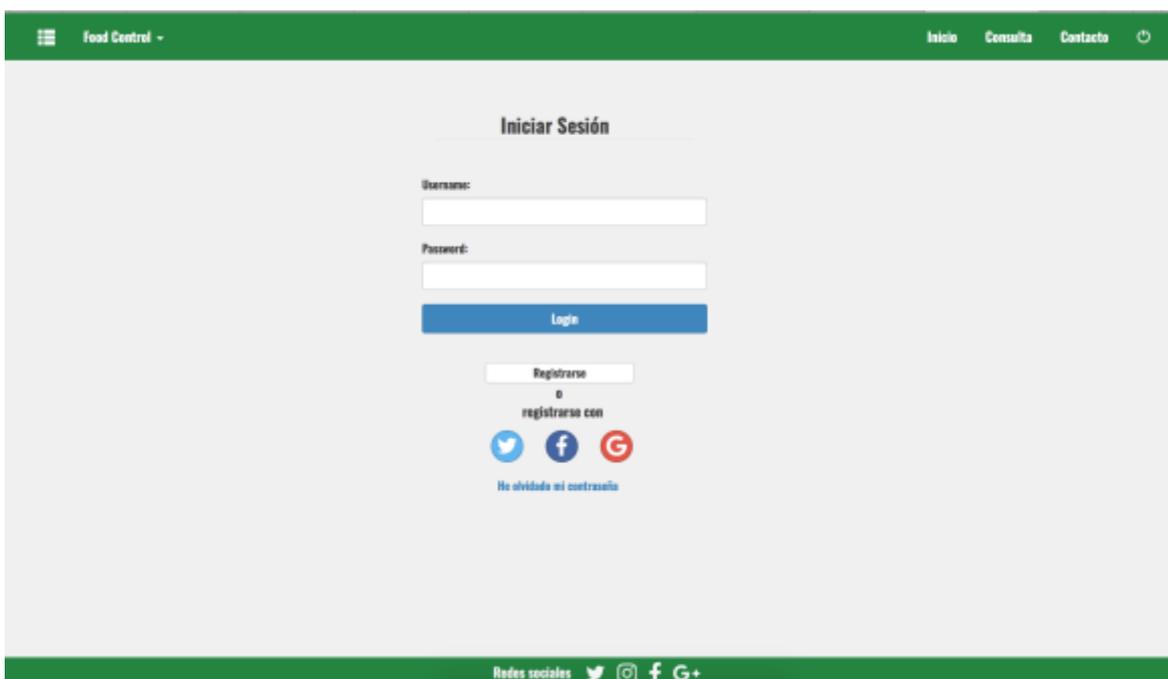
Una vez completado el perfil presionaremos continuar y estaremos en nuestro menú principal de la aplicación



Manual de usuario.



Si desea registrarse por medio de la aplicación debe pulsar el botón registrarse



Manual de usuario.

The screenshot shows the 'Alta Usuario' (User Registration) form in the 'Feed Control' application. The form includes the following fields and options:

- Nombre: [Nombre]
- Apellidos: [Apellido]
- Email: [Email]
- Username: [Username]
- Password: [Password]
- Re-Password: [Re-password]
- Doctor: [luisangelparada]
- Género: [Masculino]
- Estatura: [Estatura]
- Peso: [Peso]
- Diabetico

A 'Registrarse' button is located at the bottom of the form. The application header shows 'Feed Control' and navigation links for 'Inicio', 'Consulta', and 'Contacto'. Social media icons for Twitter, Instagram, Facebook, and Google+ are visible in the footer.

En esta vista se realizan tres validaciones

1. El nombre de usuario sea único en el sistema.

Si el nombre de usuario no es único la aplicación lo notificara y cambiara a rojo el campo de nombre de usuario.

This screenshot shows the 'Alta Usuario' form with an error message displayed. The 'Username' field, containing 'luis_a91', is highlighted in red. A modal dialog box is overlaid on the form, displaying the following information:

- Icon: A red circle with a white 'X' inside.
- Text: **Oops**
- Message: El Usuario **luis_a91** ya existe
- Button: OK

The background form shows the following values: Nombre: Alvaro, Apellidos: Lopez, Email: luis.parada@eda.usbu, Password: ****, Re-Password: ****, Doctor: luisangelparada, Género: Masculino, Estatura: 1,90, and Peso: 81. The 'Registrarse' button is visible at the bottom of the form.

Si el nombre de usuario es correcto cambiara a verde el campo de nombre de usuario

Manual de usuario.

Food Control - Inicio Consulta Contacto

Alta Usuario

Nombre:

Apellidos:

Email:

Username:

Password:

Re-Password:

Doctor:

Género:

Estatura:

Peso:

Diabetico

Redes sociales

2. Los campos de password y re-password sean iguales, si no son iguales, los campos mencionados cambian a color rojo

Food Control - Inicio Consulta Contacto

Alta Usuario

Nombre:

Apellidos:

Email:

Username:

Password:

Re-Password:

Doctor:

Género:

Estatura:

Peso:

Diabetico

Redes sociales

Una vez colocado los password iguales los campos cambiaran a color verde

Manual de usuario.

Food Control - Inicio Consulta Contacto

Alta Usuario

Nombre:

Apellidos:

Email:

Username:

Password:

Re-Password:

Doctor:

Género:

Estatura:

Peso:

Diabetico

Registrar

Redes sociales    

3. Todos los campos estén rellenos

Food Control - Inicio Consulta Contacto

Alta Usuario

Nombre:

Apellidos:

Email:

Username:

Password:

Re-Password:

Doctor:

Género:

Estatura:

Peso:

Diabetico

Registrar

Redes sociales    

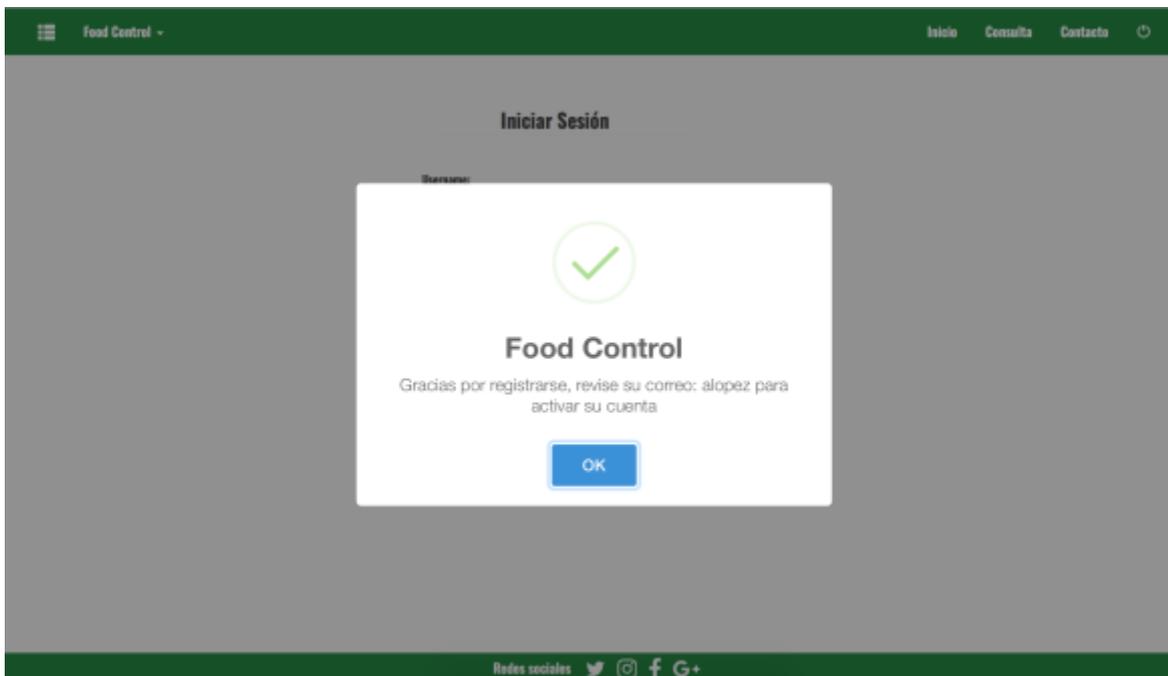
Manual de usuario.

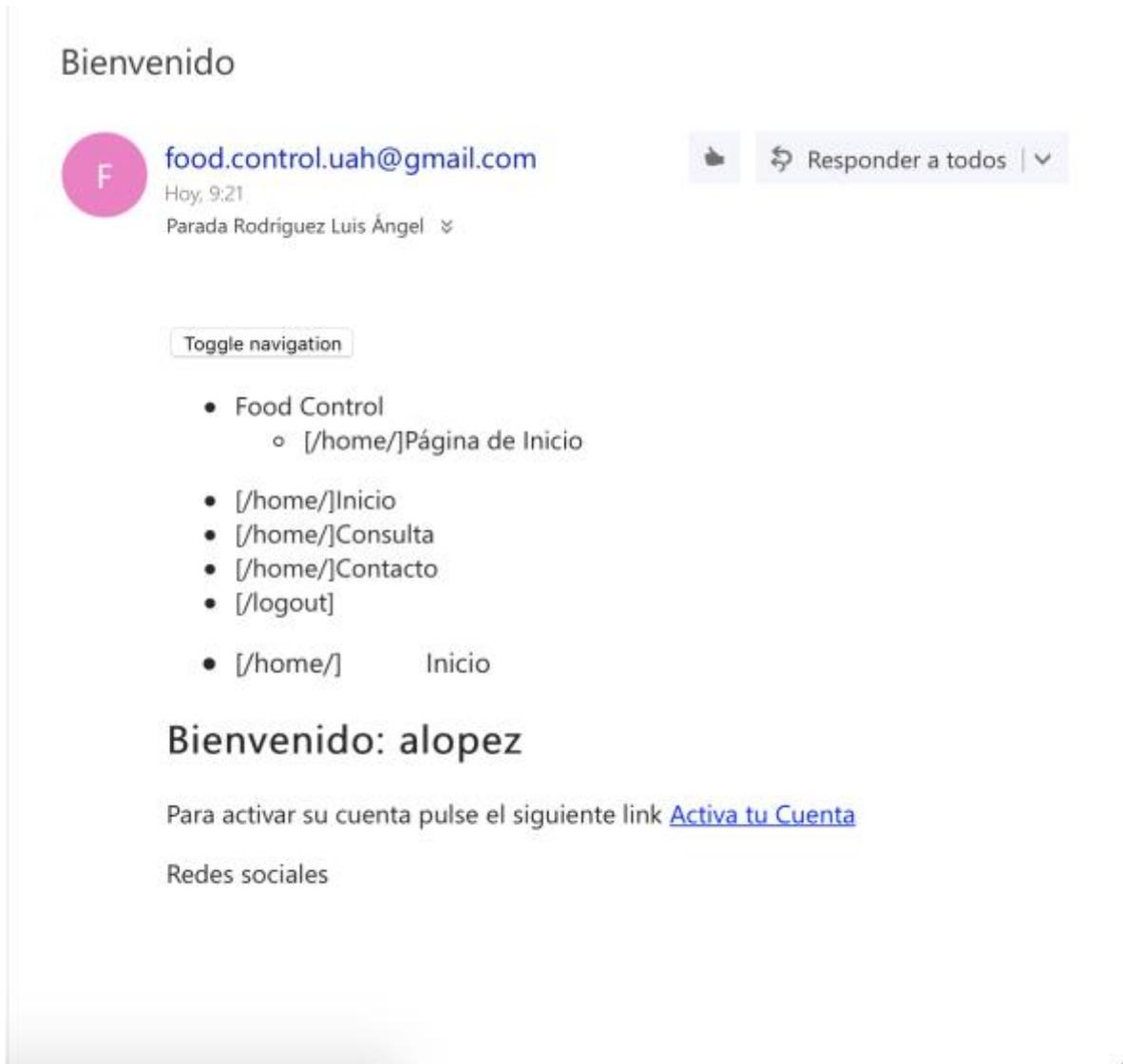
Cuando todos los campos estén rellenos se habilitara el botón registrarse que una vez pulsado, nos enviaran un correo electrónico donde debemos pinchan un link para activar nuestra cuenta

The screenshot shows the 'Alta Usuario' registration form. The form fields are as follows:

Nombre:	Alexis
Apellidos:	Lopez
Email:	lois.pareda@sis.zah.es
Username:	alopez
Password:	*****
Re-Password:	*****
Doctor:	loisangelpareda
Género:	Masculino
Estatura:	1,90
Peso:	81
<input type="checkbox"/> Diabético	

At the bottom of the form is a blue 'Registrarse' button. The top navigation bar includes 'Inicio', 'Consulta', and 'Contacto'. The footer contains social media icons for Twitter, Instagram, Facebook, and Google+.



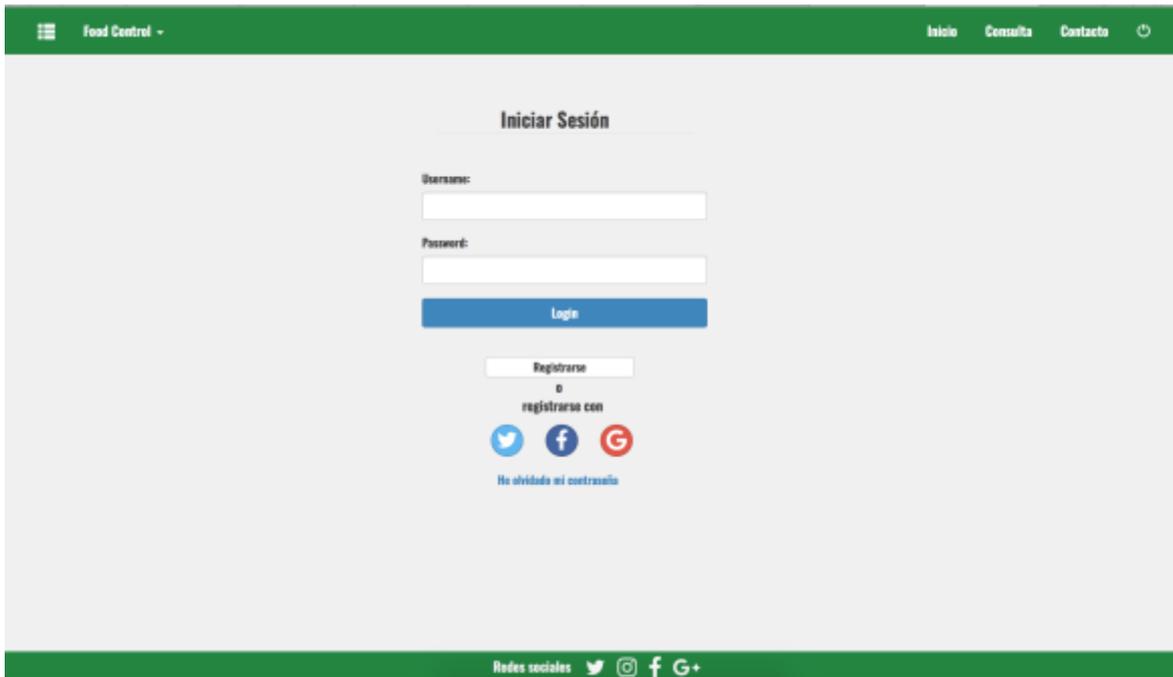


AUTENTICACIÓN

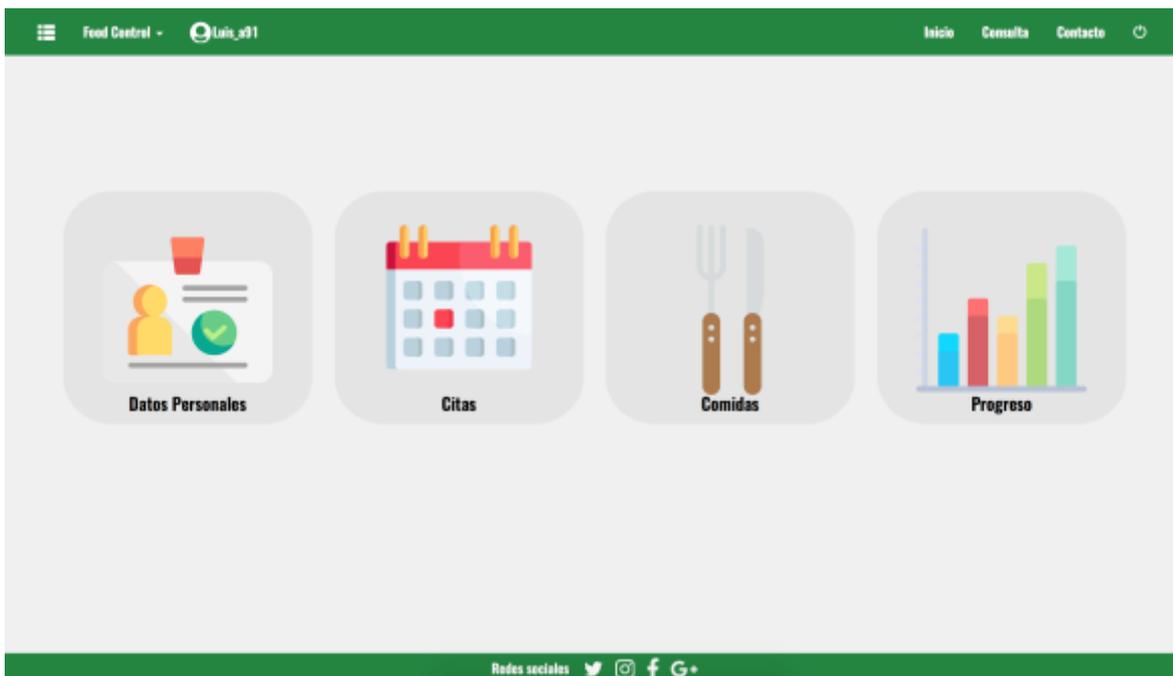
Para la autenticación también existen dos métodos por redes sociales o a través de nuestro formulario de nombre de usuario, a continuación explicaremos primero como iniciar sesión por redes sociales.

Para acceder a la aplicación a través de redes sociales solo es necesario es pinchar nuestro icono de red social con el cual nos hayamos registrado

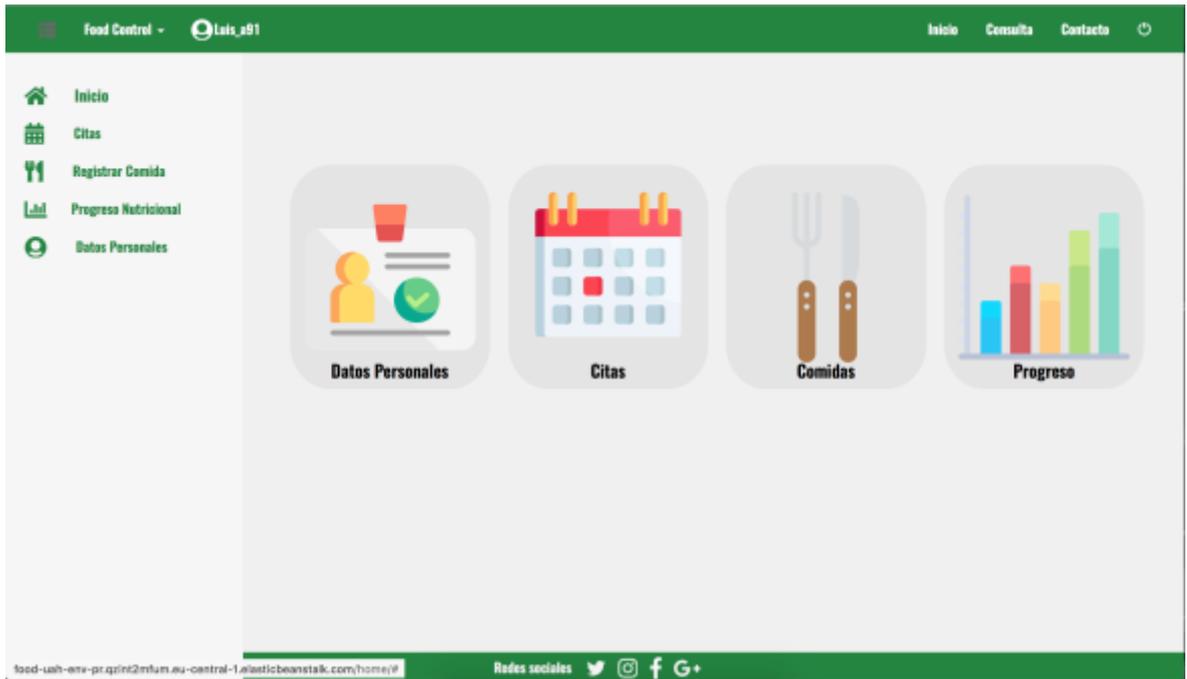
Manual de usuario.



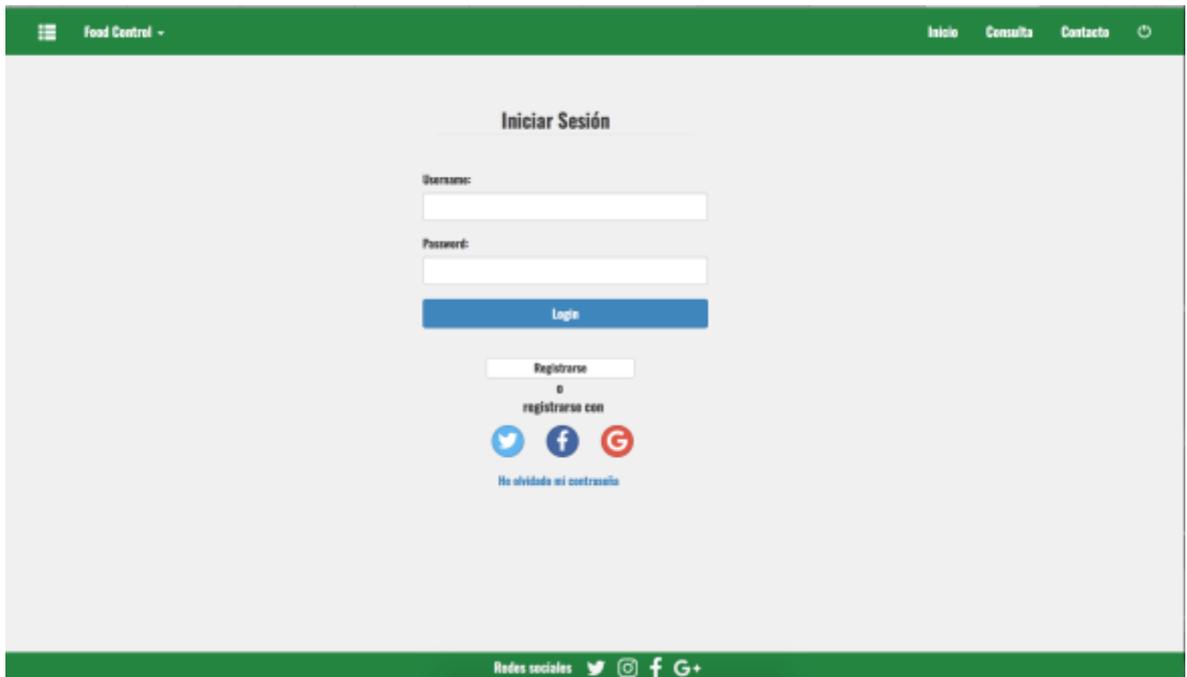
Una vez pinchado el icono y nos autentiemos con nuestra red social accederemos a la aplicación



Manual de usuario.

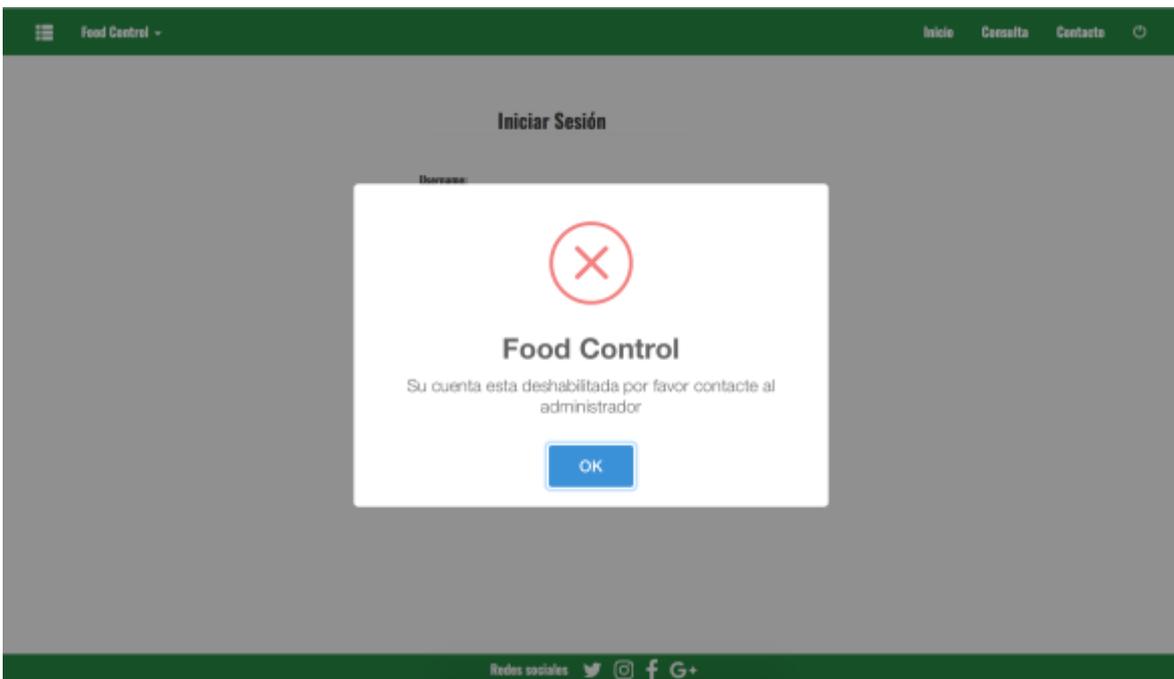
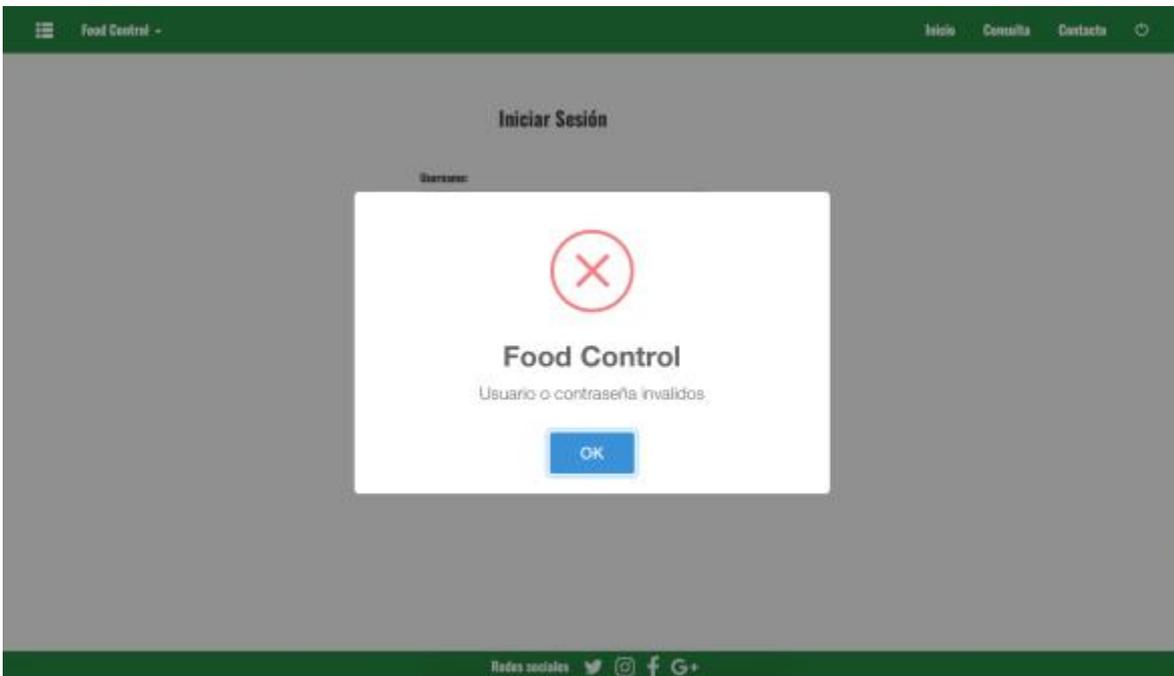


Para acceder a la aplicación a través del formulario rellenamos el nombre de usuario y el password y presionamos el botón login



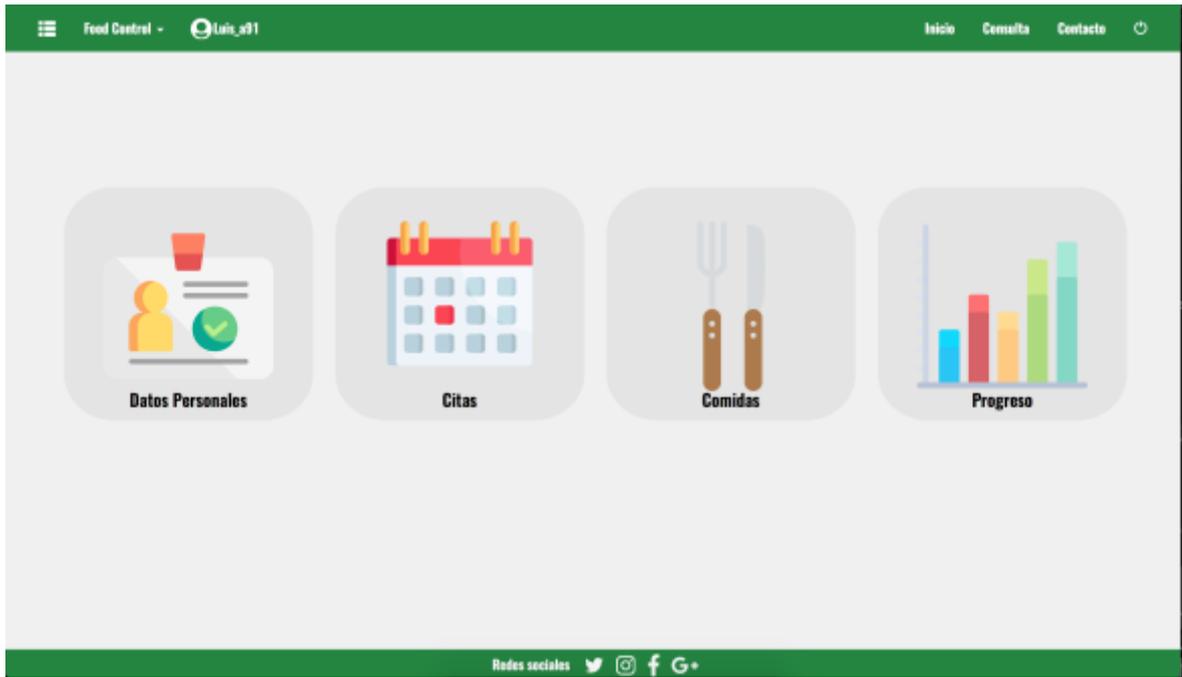
Manual de usuario.

En el login tiene dos validaciones si la contraseña o password no son correctos o que no haya activado su cuenta y le pida que contacte al administrador, igualmente la aplicación enviara otro correo electrónico para que active su cuenta.



Si el login es correcto accederemos al menú principal

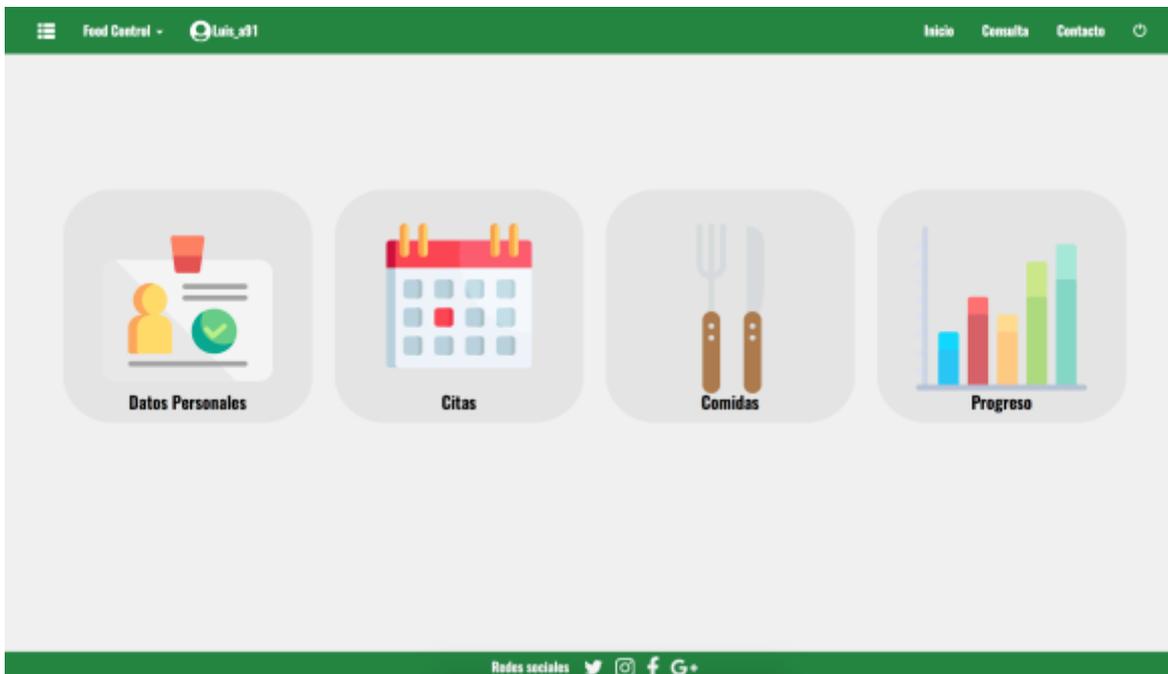
Manual de usuario.



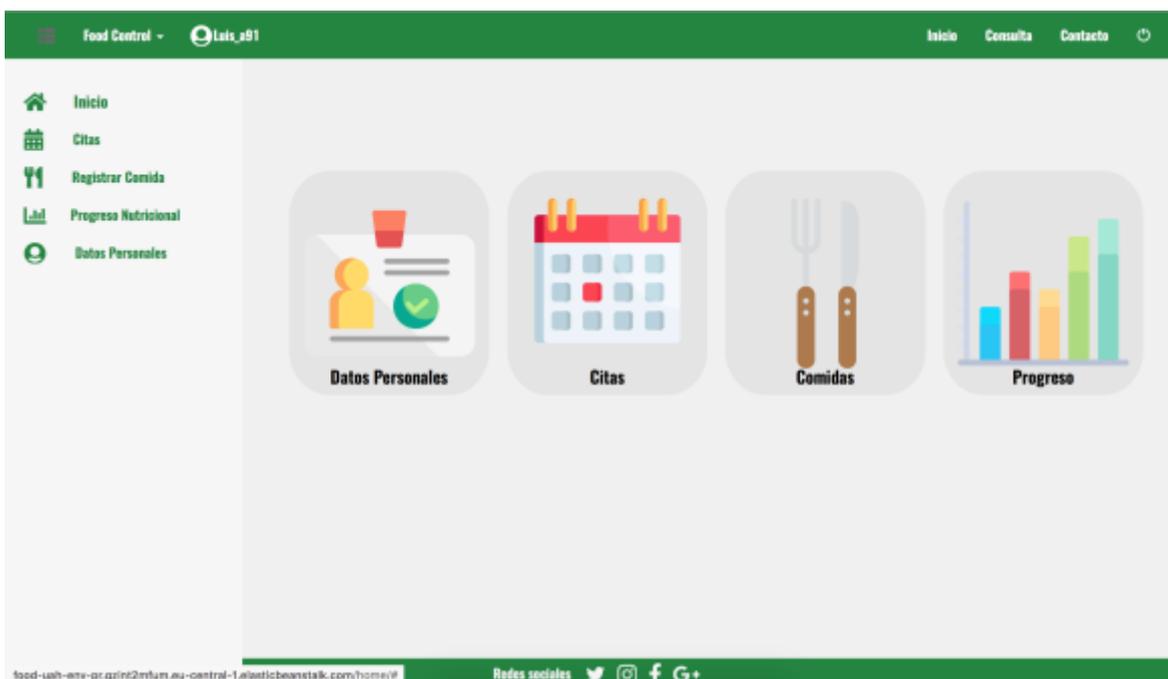
APLICACIÓN PARA EL CLIENTE

MENU PRINCIPAL

El menú principal del cliente le muestra las opciones que tiene disponible en la aplicación

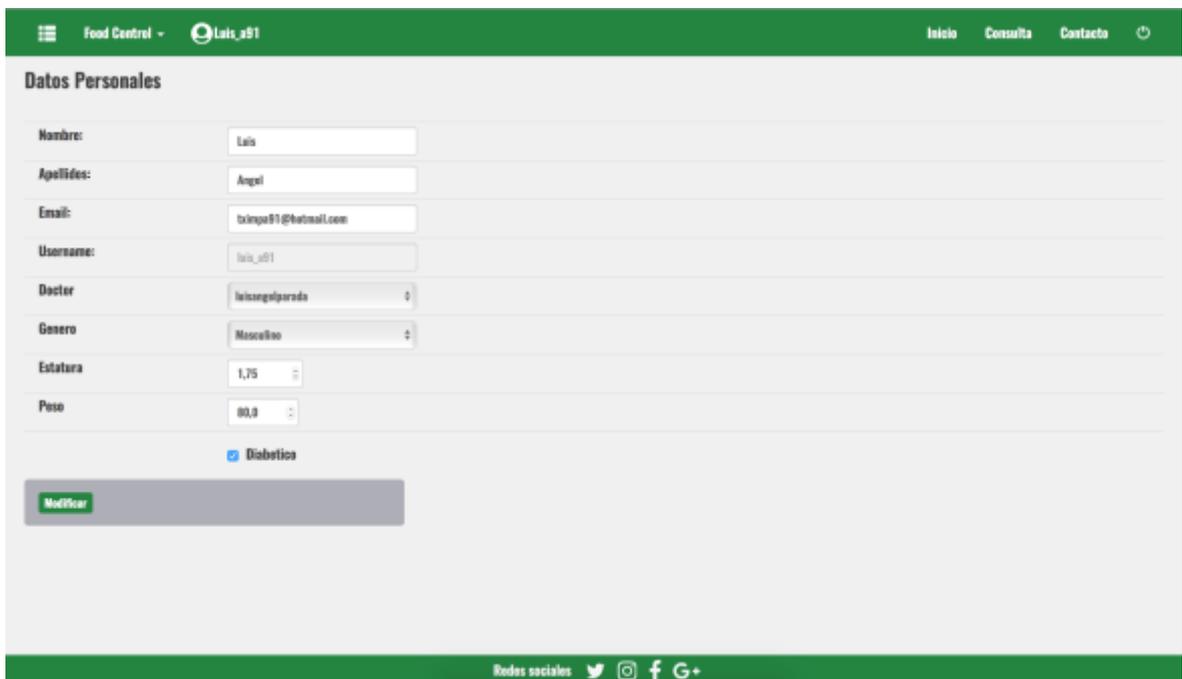


En toda la aplicación existe un botón superior izquierdo que al presionar nos despliega una barra que también nos muestra las opciones principales de la aplicación esta barra nos sirve para navegar en la aplicación.



DATOS PERSONALES

Si accedemos a la opción datos personales se mostrarán los datos personales

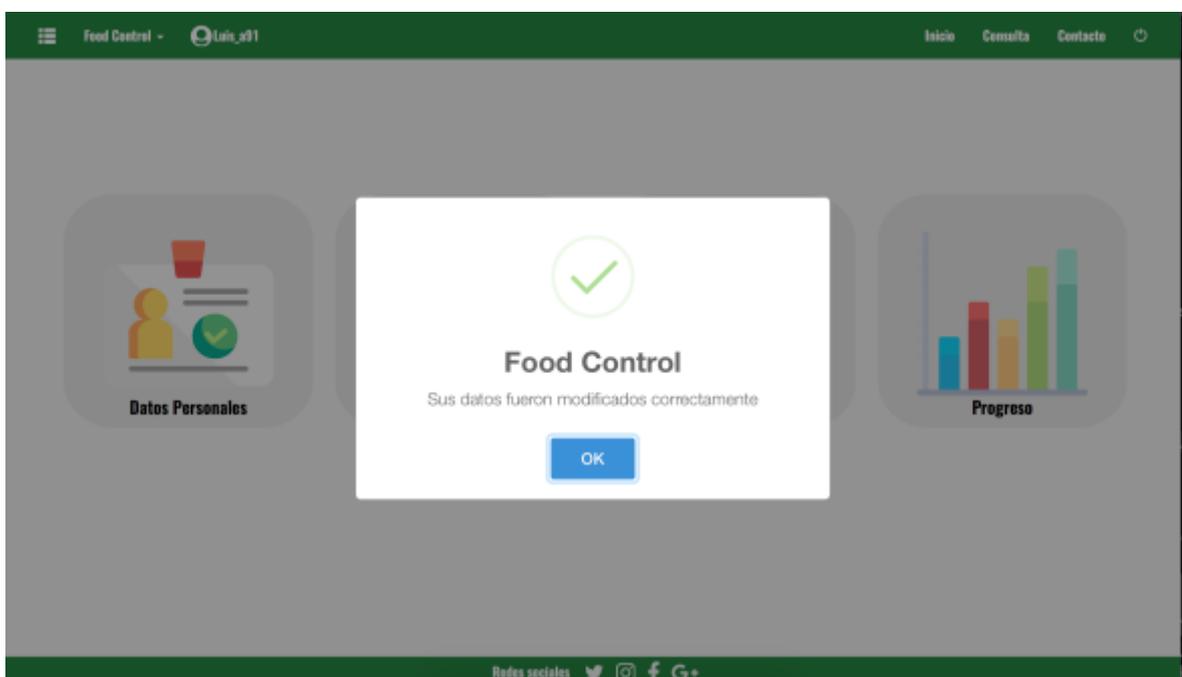


The screenshot shows the 'Datos Personales' (Personal Data) form in the Food Control application. The form is displayed on a green header with the user's name 'Luis_a91' and navigation links for 'Inicio', 'Consulta', and 'Contacto'. The form fields are as follows:

Nombre:	Luis
Apellidos:	Angel
Email:	lxmpa91@hotmail.com
Username:	luis_a91
Dokter:	Intergalparado
Genero:	Masculino
Estatura:	1,75
Peso:	80,8

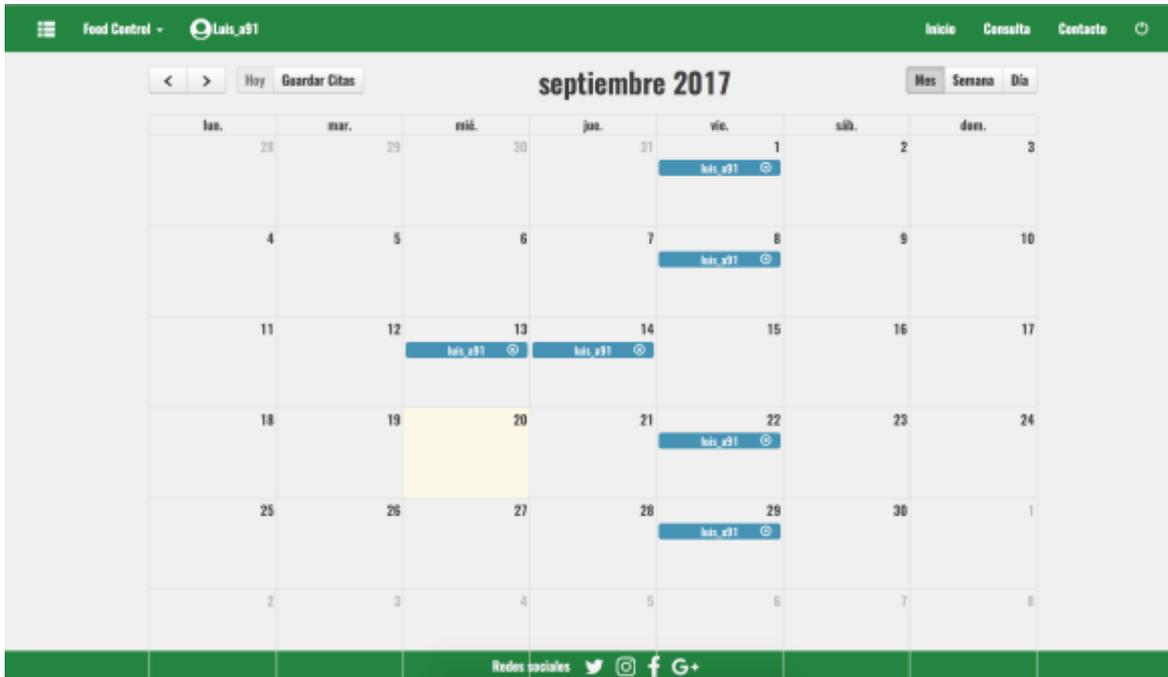
There is a checkbox for 'Diabetico' which is checked. A 'Modificar' button is located at the bottom of the form. At the bottom of the page, there are social media icons for Twitter, Instagram, Facebook, and Google+.

En esta opción podemos cambiar nuestros datos menos el nombre de usuario que es un campo único, una vez realizado los cambios se notificara el cambio e iremos al menú principal

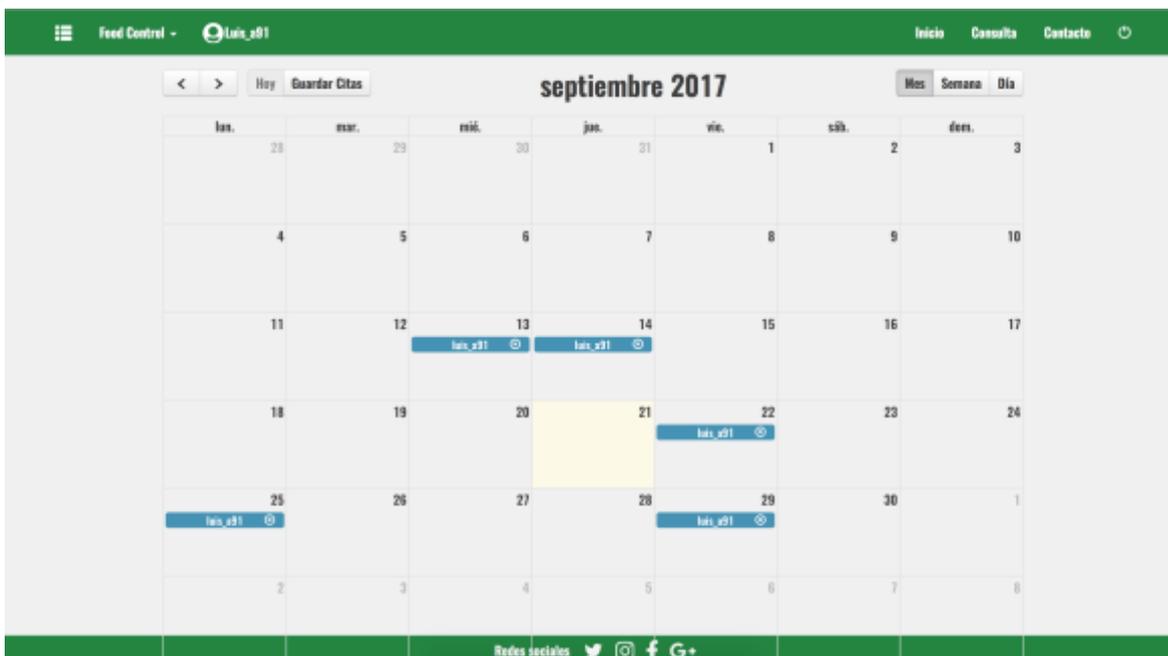


CITAS

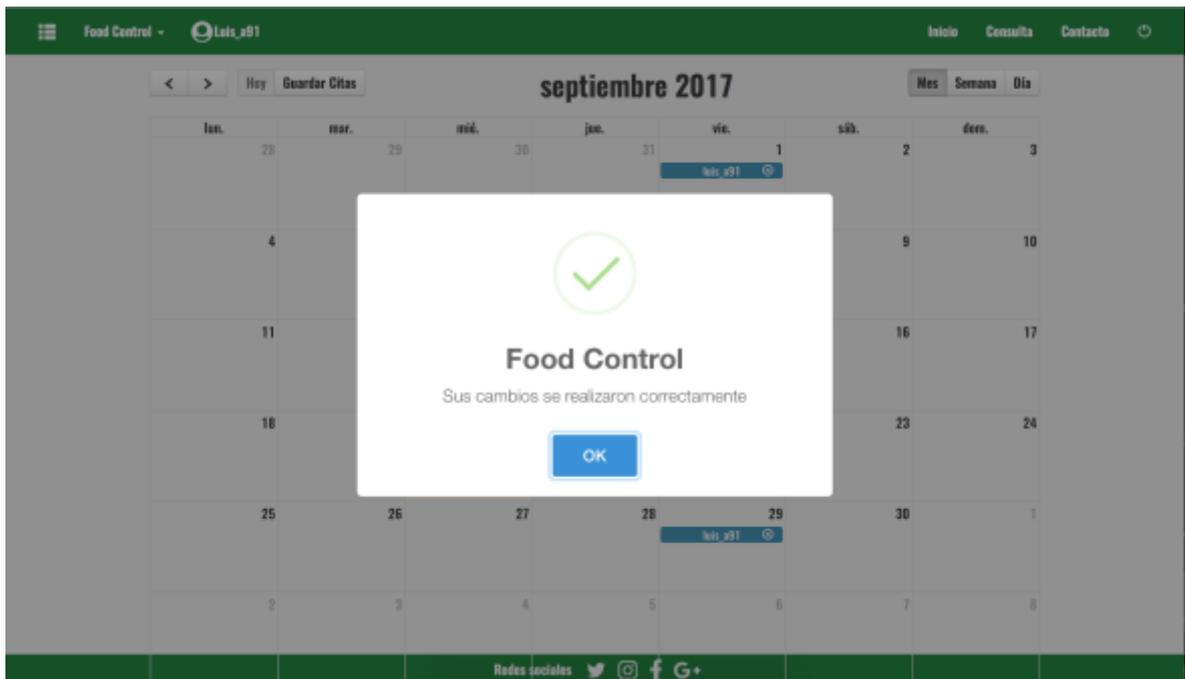
Si accedemos a esta opción la pantalla principal, tendremos un calendario situado en el mes y día en el que estemos.



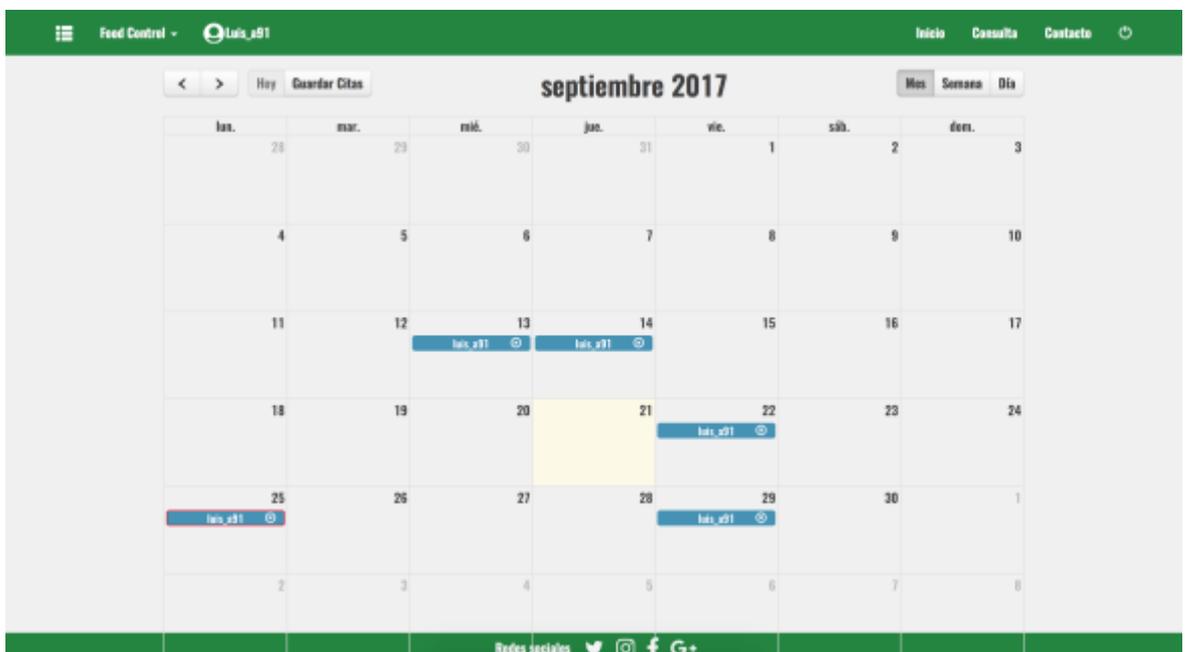
Para poder colocar una cita solo es necesario hacer un click en la fecha deseada para concertar una cita, en este caso 25 de Septiembre, para guardar los cambios presionamos el botón “Guardar Citas”



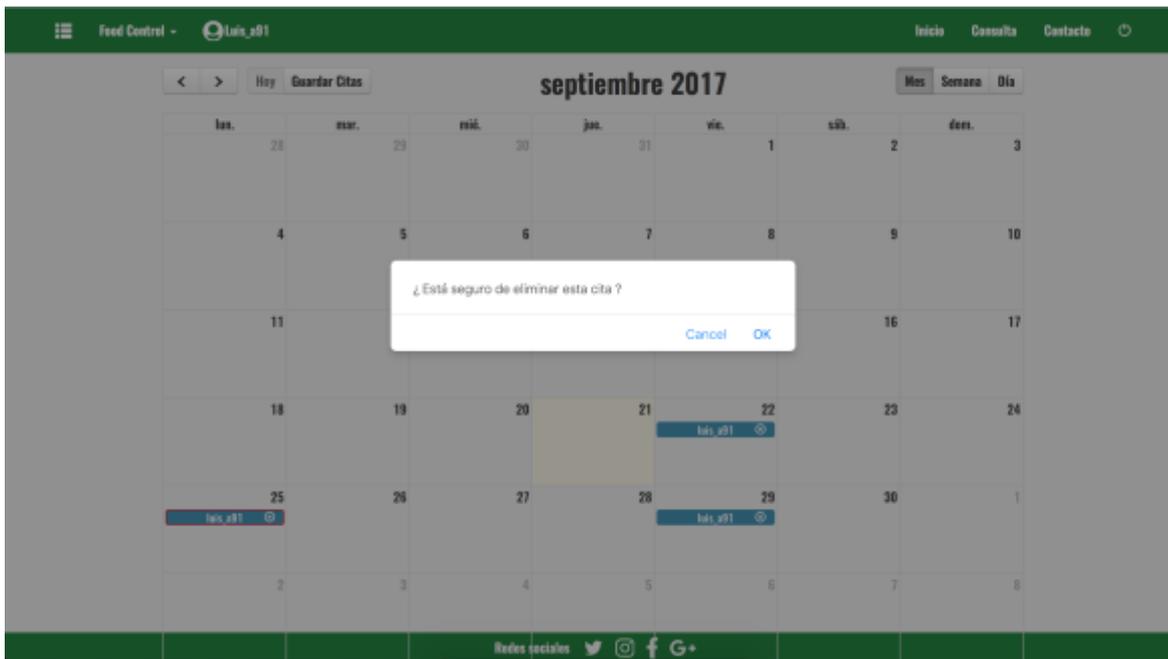
Manual de usuario.



Para eliminar una cita es necesario hacer doble clic en cada evento del calendario (rectángulo azul con nombre de usuario), el primer click colocara el borde del evento en rojo ya que ha sido seleccionado para su eliminación, al segundo click nos pedirá confirmación para eliminar esta cita, luego de presionar “Ok” es necesario presionar el botón Guardar citas. Si para ese mismo día existen más de 8 citas con ese doctor no sera posible concertar la cita para ese día

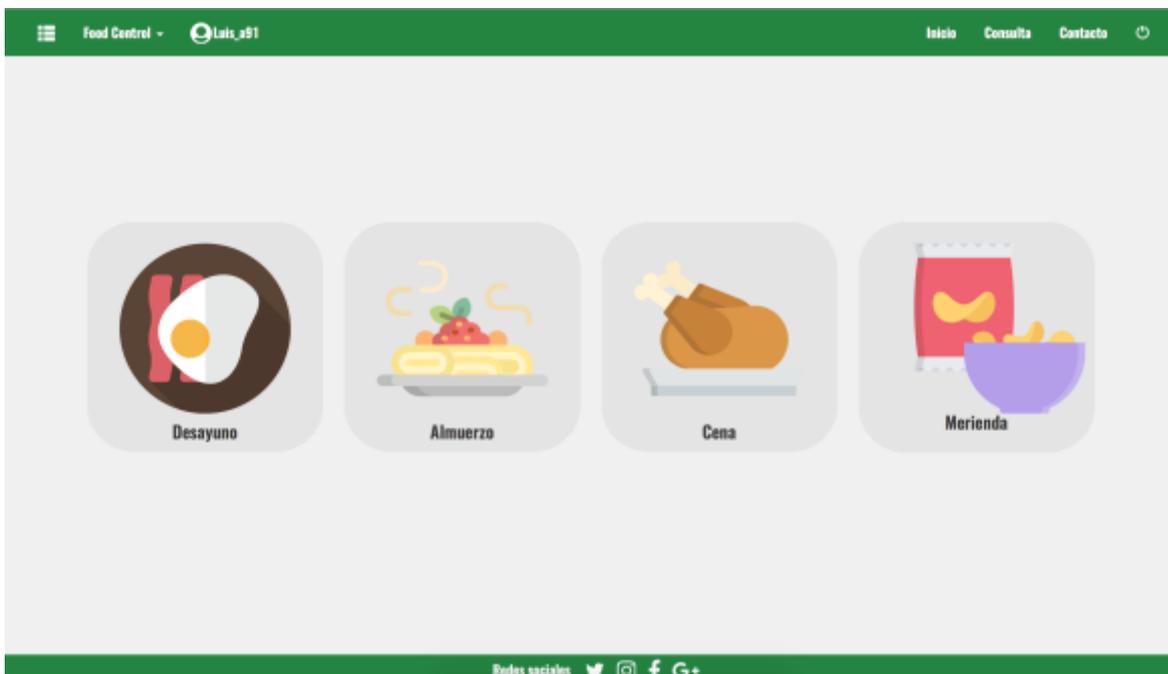


Manual de usuario.



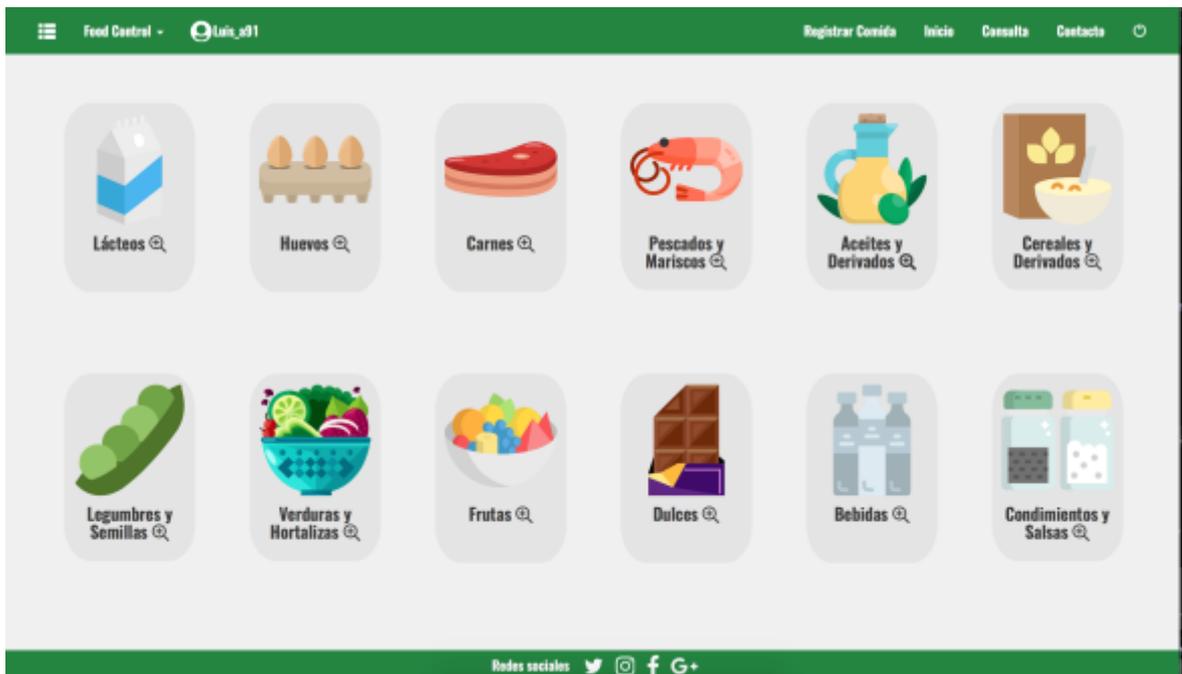
REGISTRAR COMIDA

Una vez que accedemos a esta opción tenemos 4 opciones a elegir.

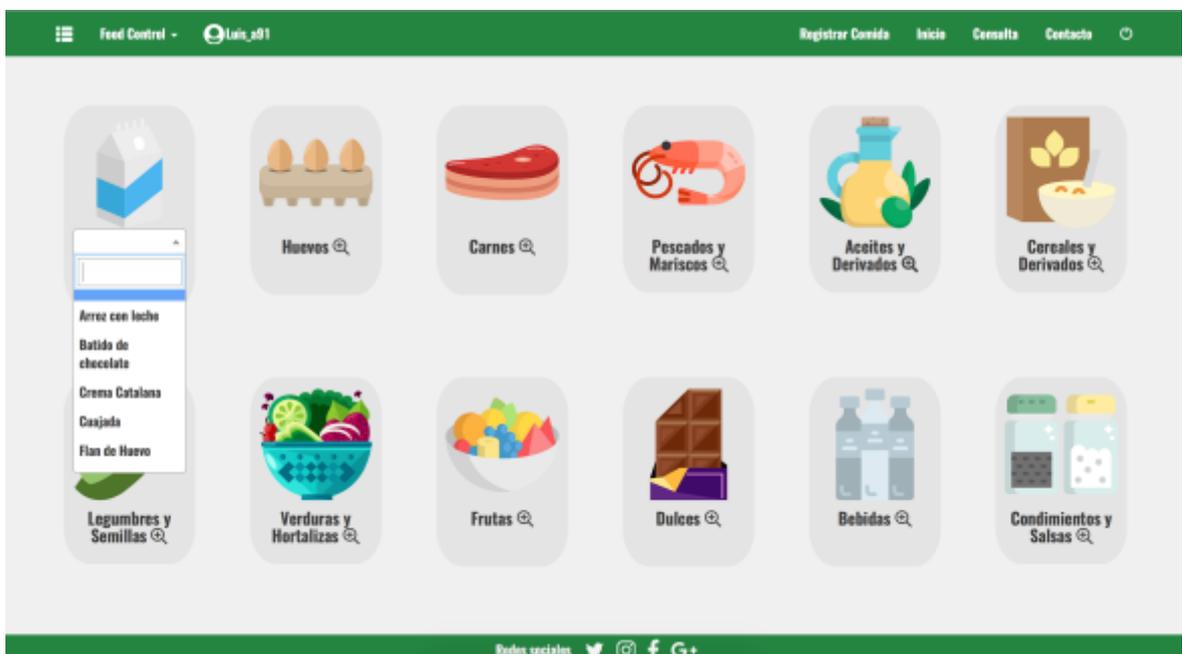


Manual de usuario.

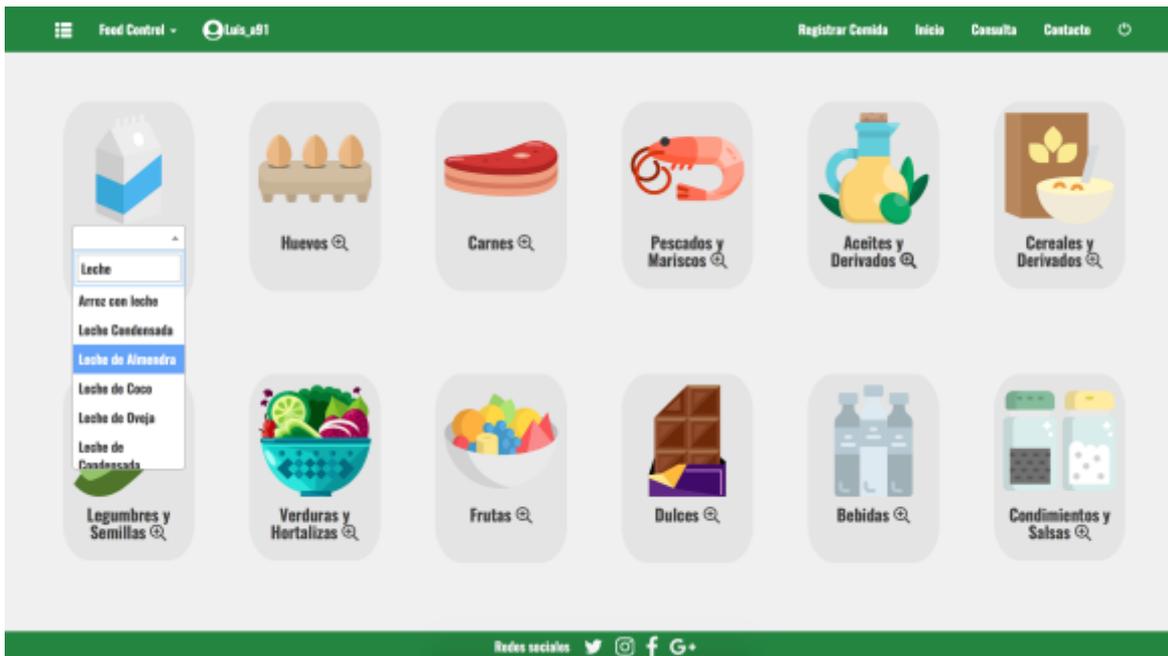
Una vez presionada esta opción tenemos este menú donde podemos elegir que alimentos elegir y rellenar la cantidad de gramos que hemos



Para desplegar la lista de alimentos de cada tipo presionamos un click en la lupa, se desplegará una barra de búsqueda donde podremos buscar el alimento que deseamos escribiendo su nombre

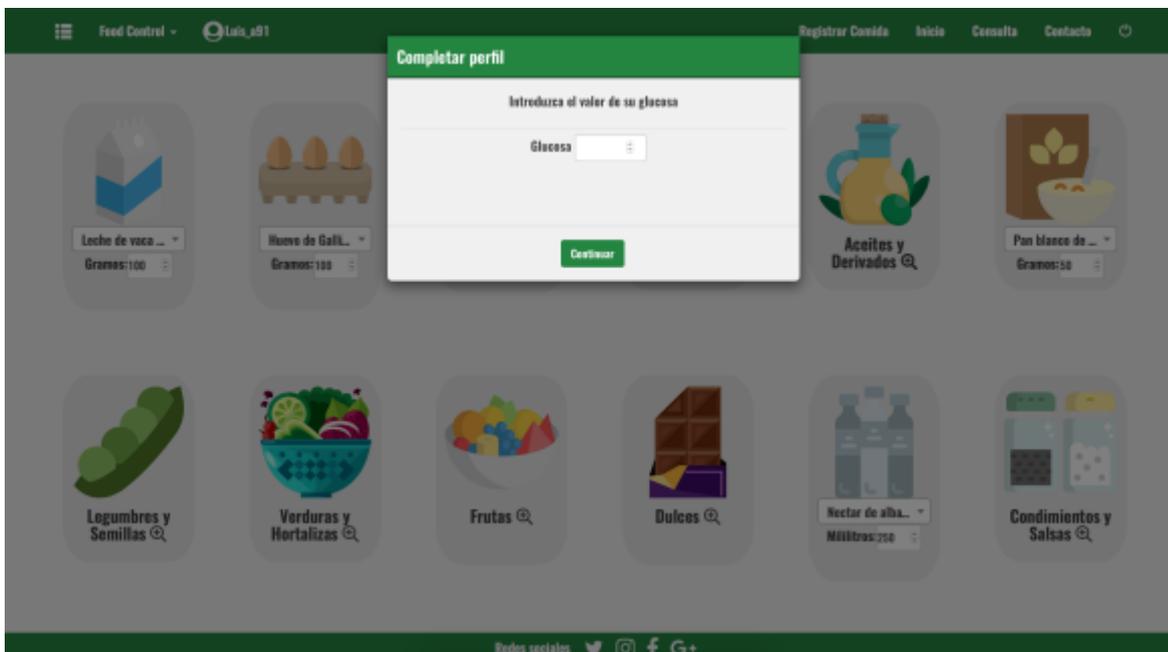


Manual de usuario.

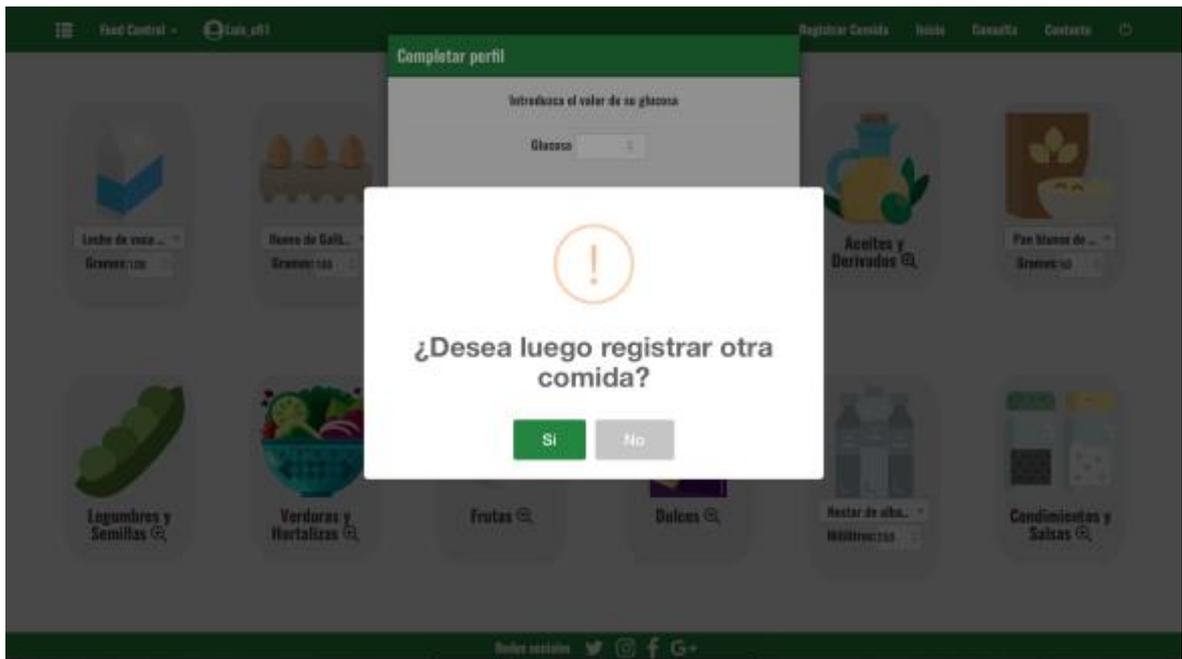


Una vez que tengamos nuestros alimentos escogidos presionamos el botón “Registrar Comida”

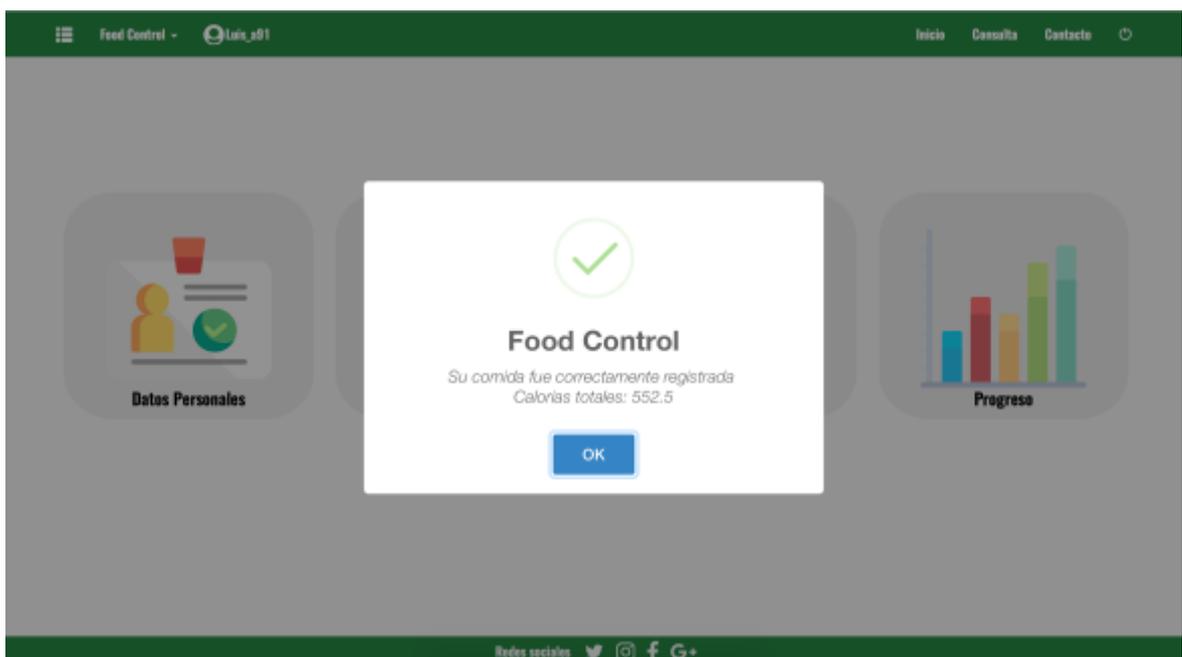
Si en nuestro perfil hemos colocado que somos diabéticos nos pedirá que introduzcamos nuestro nivel de glucosa, tendremos la opción de registrar otra comida o devolvemos al menú principal



Manual de usuario.



Luego de haber elegido una opción el sistema nos notificara que la comida fue registrada y sus calorías totales.

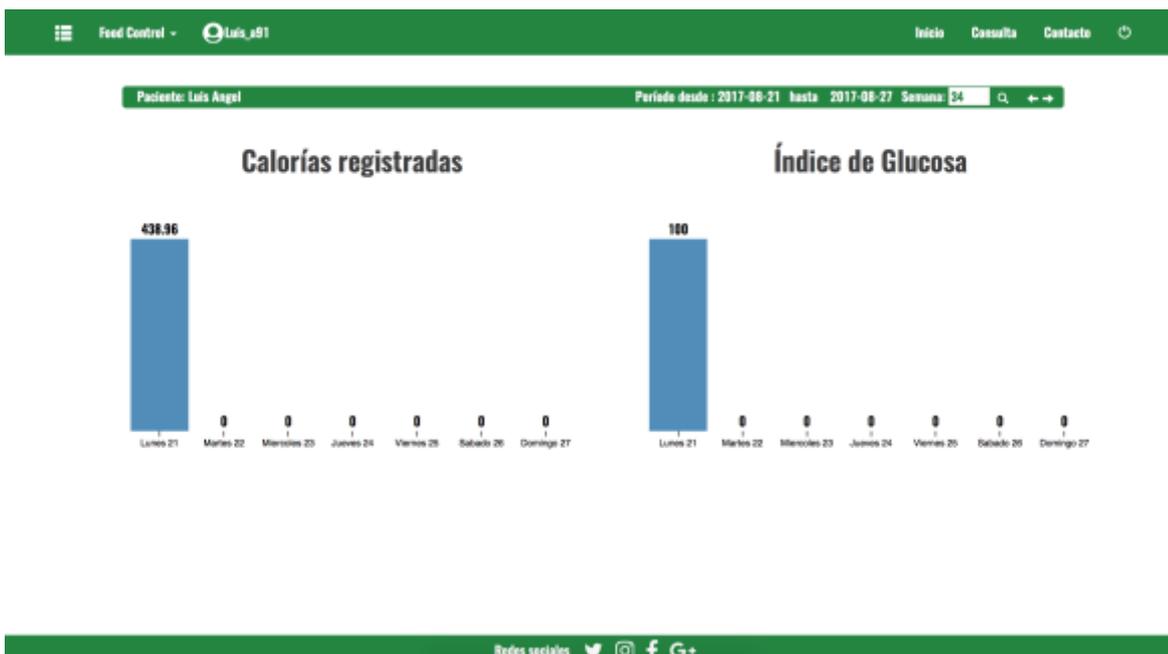


PROGRESO

Una vez que accedemos a esta opción tendremos una barra que nos indica la semana en la que estemos y tendremos un gráfico que nos informara la cantidad de comidas ingeridas por cada día de la semana, presionando las flechas izquierda retrocedemos en semana y derecha avanzamos en la semana, como también podemos escribir la semana manualmente y presionar en la lupa para ir a esa semana



Si tenemos estadísticas de niveles de glucosa para esa semana aparecerá a su lado



APLICACIÓN PARA EL NUTRICIONISTA

El nutricionista una vez que ha sido autenticado tendrá una página donde aparecerán todos los usuarios registrados con él.

Pacientes

Pacientes Registrados: 7

Username	Nombre	Apellido	Email	Peso	Estatura	Diabetico	Número de citas
luis_v91	Luis	Angel	luisangel@feedmail.com	80.0	1.75	Si	9
luisAngelP	Luis	Angel P	luisangelp@feedmail.com	70.0	1.75	Si	9
luis.parada364	Luis	Parada	luis.parada364@gmail.com	80.0	1.75	Si	0
paradalu364	Luis Angel	Parada	paradalu364@gmail.com	75.0	1.85	No	2
gabrielmanuelab46	Gabriel	Manchale	gabrielmanuelab46@gmail.com	82.0	1.71	No	0
fraguero	Fernando	Aguiroz	luis.parada@feed.usk.es	75.0	1.87	No	0
alopez	Alvaro	Lopez	luis.parada@feed.usk.es	81.0	1.9	No	0

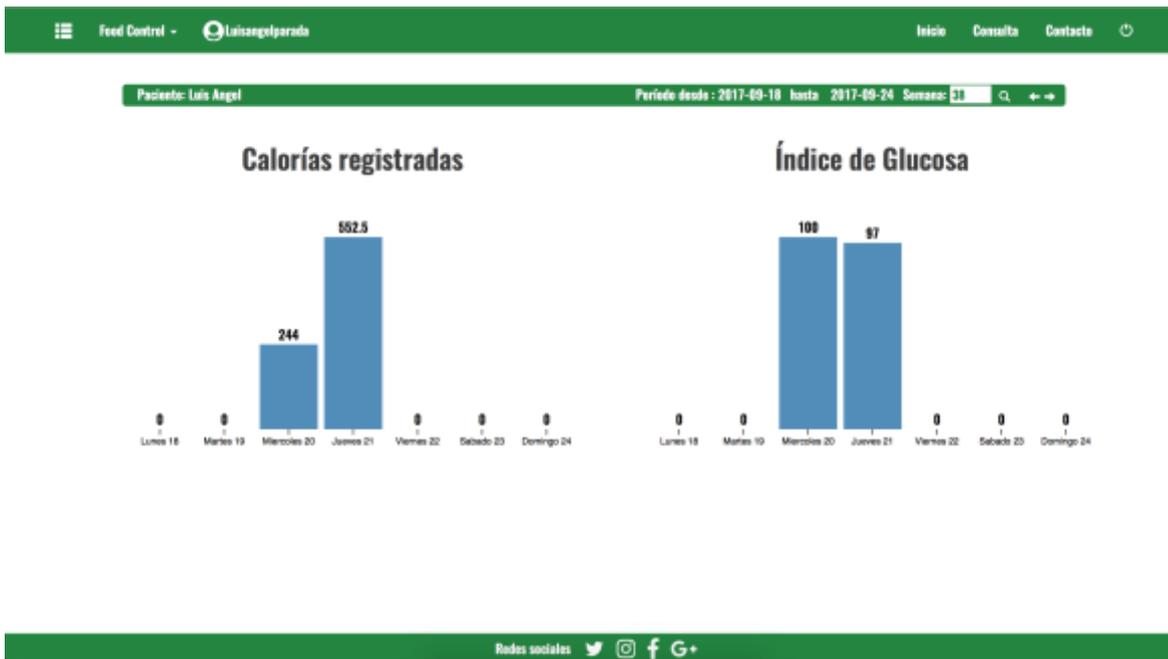
Tenemos a todos los pacientes con su información, al lado de su nombre de usuario tenemos la opción de presionar el calendario y poder concertar sus citas o presionar el gráfico y ver sus estadísticas.

Para realizar las citas es el mismo metodo que para los pacientes, la diferencia es que obtenemos la vision global de todas las citas

septiembre 2017

< > Hoy Guardar Citas
Mes Semana Día

lan.	mar.	mié.	jue.	vie.	sáb.	dom.
28	29	30	31	1	2	3
4	5	6	7	8	9	10
	LuisAngelP	LuisAngelP paradalu364	LuisAngelP paradalu364	LuisAngelP		
11	12	13	14	15	16	17
	Luis_v91 LuisAngelP	Luis_v91 LuisAngelP	LuisAngelP			
18	19	20	21	22	23	24
		LuisAngelP		Luis_v91 LuisAngelP		
25	26	27	28	29	30	1
			Luis_v91			
2	3	4	5	6	7	8



MANTENIMIENTO

El nutricionista tiene la opción de mantenimiento en la barra lateral de herramientas, donde tiene la opción de Registrar un cliente o Registrar un nuevo alimento.

Pacientes Registrados: 8

Usuario	Nombre	Apellido	Email	Peso	Estatura	Diabetes	Número de citas
luis_01	Luis	Angel	luis01@hotmail.com	80.0	1.75	Si	9
LuisAngelP	Luis	Angel P	luis01@hotmail.com	78.0	1.75	Si	9
luis.parada364	Luis	Parada	luis.parada364@gmail.com	80.0	1.75	Si	0
paradalu364	Luis Angel	Parada	paradalu364@gmail.com	75.0	1.65	No	2
gabrielaoschade6	Gabriela	Wackade	gabrielaoschade6@gmail.com	62.0	1.71	No	0
luzgalea	Fernanda	Alpino	luis.parada@luis.usb.es	75.0	1.87	No	0
alopez	Alvaro	Lopez	luis.parada@luis.usb.es	81.0	1.9	No	0
test_1	Luis Angel	Parada	luisparada364@gmail.com	90.0	1.8	No	0

Open #food-usb-env-pr.qint2refum.eu-central-1-elasticbeanstalk.com/home/# on this page in a new tab

Redes sociales: [Twitter](#) [Instagram](#) [Facebook](#) [Google+](#)

REGISTRAR CLIENTE

Funciona igualmente que un registro nuevo en el sistema a través del formulario

Manual de usuario.

The screenshot shows the 'Alta Usuario' (User Registration) form. The form fields are as follows:

Nombre:	Leis Angel
Apellidos:	Parada
Email:	leisparada2014@gmail.com
Username:	leis_1
Password:	*****
Re-Password:	*****
Doctor:	leisangelparada
Género:	Masculino
Estatura:	2,00
Peso:	90
<input type="checkbox"/> Diabotico	

At the bottom of the form is a 'Registrar' button. The page header includes 'Feed Control' and 'leisangelparada', and the footer includes 'Redes sociales' with icons for Twitter, Instagram, Facebook, and Google+.

REGISTRAR ALIMENTO

Para registrar un alimento solo es necesario rellenar todo el formulario y presionar agregar

The screenshot shows the 'Registrar Alimento' (Register Food) form. The form fields are as follows:

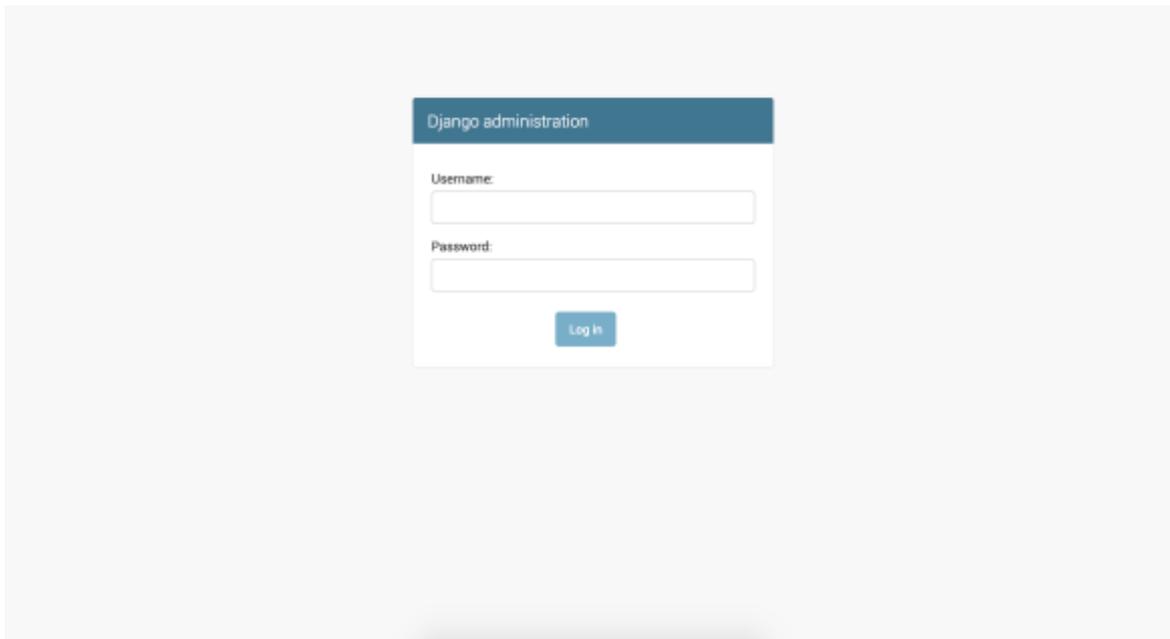
Nombre:	Nombre
Tipo:	Lacteos
Calorias:	Calorias
Grasa:	Grasa
Proteina:	Proteina
Agua:	Proteina

At the bottom of the form is an 'Agregar' button. The page header includes 'Feed Control' and 'leisangelparada', and the footer includes 'Redes sociales' with icons for Twitter, Instagram, Facebook, and Google+.

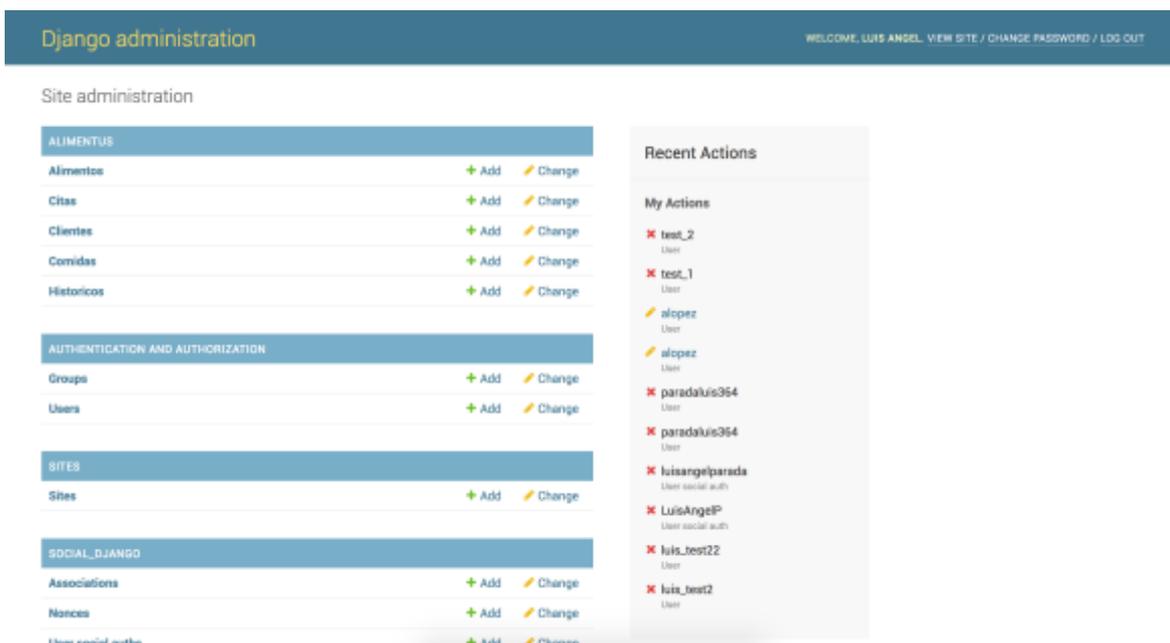
APLICACIÓN PARA EL ADMINISTRADOR

Para el/los administradores de la aplicación existe un panel de control de la aplicación generado por Django y modificado por los desarrolladores para acceder a el es necesario tener una cuenta de administrador.

Para acceder a esta funcionalidad es necesario escribir esta dirección en nuestro navegador: <http://food-uah-env-pr.qzint2mfum.eu-central-1.elasticbeanstalk.com/admin/> nos mostrara en un pantalla para iniciar sesión.



Una vez iniciada sesión podemos ver la estructura de nuestra aplicación, vemos las tablas de nuestra aplicación que se llama Alimentus, Usuarios y Autenticación, Sitios, Socia_Django que es el responsable de todo el manejo de redes sociales.



Manual de usuario.

Django administration WELCOME, LUIS ANSEL / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Alimentus

Alimentus administration

ALIMENTUS	
Alimentos	+ Add Change
Citas	+ Add Change
Clientes	+ Add Change
Comidas	+ Add Change
Historicos	+ Add Change

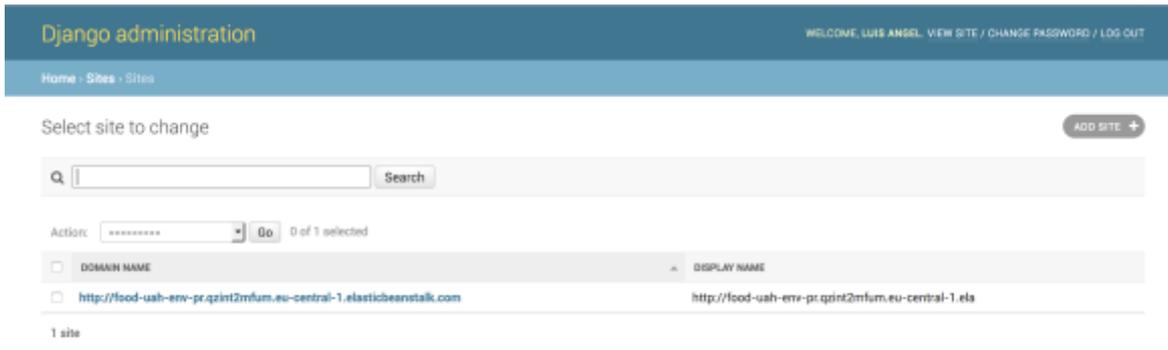
Django administration WELCOME, LUIS ANSEL / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Authentication and Authorization

Authentication and Authorization administration

AUTHENTICATION AND AUTHORIZATION	
Groups	+ Add Change
Users	+ Add Change

Manual de usuario.



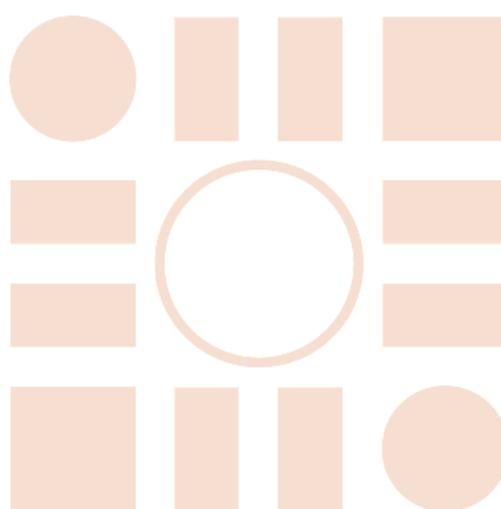
The screenshot shows the Django administration interface for the 'Sites' section. At the top, it says 'Django administration' and 'WELCOME, LUIS ANGEL. VIEW SITE / CHANGE PASSWORD / LOG OUT'. Below the header, there is a breadcrumb trail 'Home > Sites > Sites' and a title 'Select site to change' with an 'ADD SITE +' button. A search bar is present with a 'Search' button. Below the search bar, there is an 'Action:' dropdown menu and a 'Go' button, with '0 of 1 selected' indicated. A table lists the sites with columns for 'DOMAIN NAME' and 'DISPLAY NAME'. One site is listed: 'http://food-usa-env-pr.qint2mfum.eu-central-1.elasticbeanstalk.com' with a display name of 'http://food-usa-env-pr.qint2mfum.eu-central-1.ela'. At the bottom, it says '1 site'.



The screenshot shows the Django administration interface for the 'Social_Django' section. At the top, it says 'Django administration' and 'WELCOME, LUIS ANGEL. VIEW SITE / CHANGE PASSWORD / LOG OUT'. Below the header, there is a breadcrumb trail 'Home > Social_Django' and a title 'Social_Django administration'. A table lists the objects with columns for 'SOCIAL_DJANGO', 'Associations', 'Names', and 'User social auths'. Each row has '+ Add' and 'Change' buttons.

En cada una de las opciones podemos consultar, agregar, eliminar objetos de respectiva tabla a elegir

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá