



CIT@MEDICA

**Decisiones  
Tomadas**

**CIT@MEDICA**

**Fecha: 24/11/2022**

## Tabla de contenido

Tabla de versionado .....	2
Base de datos .....	3
Página de inicio .....	4
Proceso de compra .....	5
Cesta de la compra .....	5
Administrador .....	6
Incidencias .....	6
Solicitudes de cambio.....	7
Herramientas a utilizar .....	7

## Tabla de versionado

Versión	Modificado por	Descripción	Fecha
<b>V0.1</b>	Miguel Gaviro Martínez	<i>Versión inicial del documento</i>	14/11/2022
<b>V0.2</b>	Jose M <sup>a</sup> García Quijada	<i>Versión de la iteración 1</i>	20/11/2022
<b>V0.3</b>	Jose M <sup>a</sup> García Quijada	<i>Versión de la iteración 2</i>	24/11/2022
<b>V0.4</b>	Jose M <sup>a</sup> García Quijada	<i>Versión de la iteración 3</i>	24/11/2022

## Iteración 0

A continuación, se describirán las decisiones tomadas en el proyecto para la elaboración del mismo en la iteración 0 o iteración de partida.

## Base de datos

La primera decisión en cuanto a la base de datos fue elegir **MySQL** como sistema de gestión de bases de datos relacional.

Después, elegimos cómo haríamos cada tabla del modelo:

### Citas

Atributo	Restricciones
ID	auto-increment, not null, unique
Fecha y hora	datetime, not null
Nombre del cliente	String, not null
Apellidos del cliente	String, not null
Correo Electrónico	Email, not null
Precio	Double, not null
Médico	ManyToOne

- UNIQUE(ID, Fecha y hora, Médico)
- UNIQUE(Departamento, correo electrónico)
- La fecha debe de ser como mínimo 3 días después de la fecha actual en la que se pide la cita.

### Departamento

Atributo	Restricciones
Nombre	primary key, string

Atributo	Restricciones
Número de médicos	Derivada.

## Médico

Atributo	Restricciones
Nombre	String, not null
Apellidos	String, not null
Departamento	ManyToOne

## Cliente

Atributo	Restricciones
ID	primary key, integer, auto increment
Correo Electrónico	notnull, unique, email
Nombre	String, not null
Apellidos	String, not null
DNI	Regex(Formato español)
Género	Enum(Masculino, Femenino, Otro), not null
Nacionalidad	String, not null
Citas	OneToMany, puede estar vacío

## Página de inicio

Se ha decidido que habrá tres tipos de vista diferentes de página de inicio:

- Página de inicio Sin Iniciar Sesión.
- Página de inicio Usuario Registrado.
- Página de inicio Administrador.

## Proceso de compra

En el proceso de compra se podrá buscar una cita tanto por departamento como por médico. Siendo una lista de citas filtrada por ambas selecciones.

Si selecciona un **departamento**, deberá de mostrarse su nombre y un botón para pedir una cita para ese departamento

Si busca por **médico**, igual pero saldrán las citas disponibles de ese médico en el calendario y al pulsar en un día, se mostrarán las franjas horarias que ese médico tiene.

En cuanto al catálogo:

- Se mostrarán todos los departamentos disponibles con un botón que permita seleccionarlos. Entonces deberá de mostrarse un calendario donde el cliente pueda elegir una fecha, y que al elegirla se le desplieguen las franjas horarias disponibles junto al médico que atenderá la cita.
- Los médicos que salen deberán ser los pertenecientes al departamento seleccionado.
- Tras seleccionar una franja horaria, volver a la vista de elegir departamentos, pero marcando la opción o las opciones elegidas antes como "seleccionadas".
- Debe de poderse visualizar el estado de la cesta de la compra y volver.

## Cesta de la compra

- Debe de ser visible en todo momento como cliente.
- Se deben de poder quitar citas de la cesta de la compra
- Las citas sólo se añaden desde el catálogo.

- Deberá de mostrarse una lista con las citas que se han seleccionado informando del departamento, médico, fecha y franja horaria en la que se seleccionó.
- Al final de la lista se especificará la suma total de los precios de cada una de las citas.

## Administrador

Debe mostrarse un listado en el que se muestren las **citas** y pueda buscar una según su ID.

Deben de poder realizarse todas las operaciones CRUD con las citas, esto incluye:

- **Create:** Crear citas asociadas a un usuario y a un médico, con su franja horaria y fechas asignadas.
- **Update:** Actualizar el médico y la franja horaria de una cita. Nunca modificando al usuario que la pidió.
- **Delete:** Cancelar una cita.

De esta misma forma que puede gestionar citas, podrá gestionar **clientes** y **médicos**.

## Incidencias

Los **clientes registrados** podrán crear incidencias sobre citas si lo ven necesario.

- Se debe poder elegir el tipo de incidencia: Anular cita, cambio de mi información, otros.
- El cliente debe aportar una pequeña descripción con los motivos de la incidencia.

El **administrador** debe de tener acceso a las Incidencias creadas, para visualizarlas y cerrarlas. Al cerrar una incidencia se supondrá que queda archivada, pero no se borrará del sistema.

## Solicitudes de cambio

Hay varias solicitudes de cambio en nuestro proyecto, pero solo hay dos aceptadas hasta el momento, que se llevarán a cabo:

- **Reestructurar los requisitos:** Reestructurar todos los requisitos para darles sentido a la hora de desarrollarlos, incluyendo actividades para mockups, organizándolos por iteraciones y por grupos de actividades...
- **Cambios en el presupuesto:** Los ordenadores debemos de poner su coste en el presupuesto, así como todo coste personal que hayamos obviado.

## Herramientas a utilizar

Se ha decidido que cada uno usará su propio IDE para realizar el proyecto, de esta forma será más cómodo y eficaz para cada miembro.

También cabe destacar que se ha decidido que se usará una plantilla de Django para agilizar la creación de la web ya que no se dispone de mucho tiempo.

## **Iteración 1**

Tras ver las decisiones tomadas en la iteración 0 para el desarrollo del proyecto, pasamos a describir cuáles han sido las decisiones que han sido relevantes tras el desarrollo de esta primera iteración. Estas nacen a partir de una necesidad originada por las decisiones anteriormente tomadas o por no ser necesaria tomar dicha decisión con anterioridad.

### **Base de datos actual**

Al comenzar la iteración 0, decidimos que sería buena idea implementar una base de datos MySQL en nuestro proyecto de Django, no obstante, a la hora de realizarlo nos dimos cuenta de que era más fácil y útil implementar una base de datos SQLite ya que esta se adaptaba de una mejor forma a nuestro proyecto y forma de trabajar.

### **Uso de documentación y plantilla**

A la hora de desarrollar la aplicación, puesto que no contábamos con experiencia previa con Django, decidimos crear la página web desde 0 siguiendo la documentación de Django.

Pese a aprender cómo funcionaba y obtener grandes conocimientos sobre el tema, decidimos que para economizar tiempo y recursos sería óptimo plantear la idea de recurrir a una pequeña plantilla que nos marque el camino para el correcto desarrollo de la web.

Finalmente optamos por el uso de la plantilla, la cual, tuvimos que adaptar en numerosos ámbitos para que verdaderamente fuese útil en nuestro caso.



## **Iteración 2**

Una vez expuestos los problemas que tuvimos a la hora de desarrollar la iteración 1 y los diferentes retrasos que vivimos durante su desarrollo, tomamos diferentes decisiones al respecto para revertir dicha situación. Estas decisiones tomadas en la iteración 2 nacen a partir de una necesidad originada por las decisiones anteriormente tomadas o por no ser necesaria tomar dicha decisión con anterioridad.

### **Ceñirnos estrictamente a lo necesario**

El equipo de trabajo tenía pensado implementar un sistema de registro y de inicio de sesión en la página además de pequeñas funcionalidades en la página web, no obstante, hemos decidido que la mejor opción a la hora de desarrollar el proyecto es ceñirnos a lo necesario para cumplir con los plazos de entrega, ya que el equipo no dispone del tiempo necesario para el desarrollo de estas tareas que no son estrictamente necesarias para cumplimentar el proyecto.

### **Redistribuir la organización de las tareas**

Puesto que, como ya hemos hablado antes, tuvimos un retraso en la primera iteración, hemos decidido acomodar y reestructurar la organización de las tareas en los siguientes entregables de forma que el equipo pueda trabajar de forma adecuada pese a contar para dichos entregables con una mayor carga de trabajo.

## **Iteración 3**

Tras ver las diferentes decisiones tomadas en las iteraciones anteriores y los correspondientes problemas que pretendían mitigar, vamos a ver, tras volver a un orden natural de proyecto sin retrasos que problemas nos hemos encontrado en esta iteración. Estas decisiones tomadas en la iteración 3 nacen a partir de una necesidad originada por las decisiones anteriormente tomadas o por no ser necesaria tomar dicha decisión con anterioridad.

## **Cambio de plantilla de trabajo**

Después de tratar con una plantilla que parecía que podía encajar en nuestro proyecto adaptándola, nos dimos cuenta de que finalmente dicha plantilla era más sofisticada y compleja de lo que nosotros en un principio necesitamos para lograr nuestros objetivos por lo que terminamos tomando la decisión de desechar dicha plantilla y empezar a tratar con otra que encajase de forma certera con nuestro objetivo de crear una página de citas médicas.

## **Integración de Calendario Django**

Puesto que nuestra página se encarga de ofrecer citas médicas en un intervalo de tiempo, es necesario un calendario donde el usuario pueda elegir el día que mejor se adapte a sus necesidades. Para este requisito y todos los relacionados con él, hemos implementado un calendario que cumple a la perfección con aquello que necesitamos, aunque serán necesarias diversas modificaciones en él.

## **Uso del panel de control del Administrador de Django**

Tras debatirlo entre los miembros del grupo decidimos, con el objetivo de economizar tiempo de desarrollo, utilizar el panel de control del administrador que viene por defecto en el framework de Django ya que consideramos que este es fácil de usar y muy intuitivo para el usuario, aunque no se tengan conocimientos previos de Django.

