

**EVIDENCIA CALIFICABLE #**

Nombre de la Unidad de Aprendizaje : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Nombre del proyecto: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Programa educativo: Licenciado en Tecnologías de Información\_\_\_\_\_\_\_\_\_\_

Semestre: \_\_\_\_ Grupo: \_\_\_\_\_\_

Nombre del maestro: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Nombre de los integrantes del equipo:

(En orden alfabético apellidos-nombres- # matrícula)

|  |  |
| --- | --- |
| **Contenido mínimo a evaluar** | **Cumplimiento** |
| Índice |  |
| Introducción .- incluye valores UANL aplicados |  |
| Análisis y emisión de juicio |  |
| Conclusiones individuales |  |
| Conclusión del equipo |  |
| Identificación de sub resultados de aprendizaje ANECA. |  |
| Calificación Evidencia: |  |
| Firma del maestro | |

San Nicolás de los Garza, ciudad universitaria a (día-mes-año)

INDICE

Objetivo 3

APIS utilizadas 3

Google Calendar 3

Tecnologías 5

Node.js 5

Características principales de Node.js 7

Express.js 7

Medios utilizados 8

Medios hardware 8

Medios software 8

Resultados 9

Diseño de la interfaz 11

Vista de inicio 11

Implementación 12

Conclusiones 14

Mejoras futuras 15

ANEXOS 16

Manual de usuario de la aplicación 20

Vista de inicio 22

Alta de reserva 23

Selección de participantes, fechas, horas y salas 25

Formulario de datos de la reserva 28

PSEUDOCODIGO 30

Importación de Calendario: 35

## Objetivo

**Objetivo principal**

El objetivo del presente [trabajo](#_bookmark7) es desarrollar una aplicación que ayude en las tareas relacionadas con la gestión de reserva de salas y tiempo de trabajo, así como almacenar un histórico tanto de las reservas que se harán en el día a día como de las reuniones ya finalizadas.

Para alcanzar el objetivo principal, primero se deberán conseguir los siguientes objetivos parciales:

Desarrollo de un módulo de comunicaciones entre la aplicación y la [BD](#_bookmark7).

Desarrollo de un módulo de comunicaciones cliente-servidor.

Desarrollo del cliente de la aplicación.

## APIS utilizadas

### Google Calendar

Google Calendar es una aplicación que forma parte del entorno *GSuite* de Google. Esta aplicación es gratuita siempre y cuando sea para uso personal, pero en caso de ser para uso de empresa, la empresa deberá realizar una suscripción de pago. Esta aplicación implementa un calendario interactivo que nos permite realizar infinidad de acciones sobre él.



Entre las funcionalidades que tiene esta aplicación, destacan las siguientes: Programar, editar y eliminar eventos

Invitar a personas a los eventos programados. Reserva de salas y recursos.

Añadir información adicional a eventos, así como adjuntar ficheros en ellos. Notificar a los invitados de la creación y modificación de eventos.

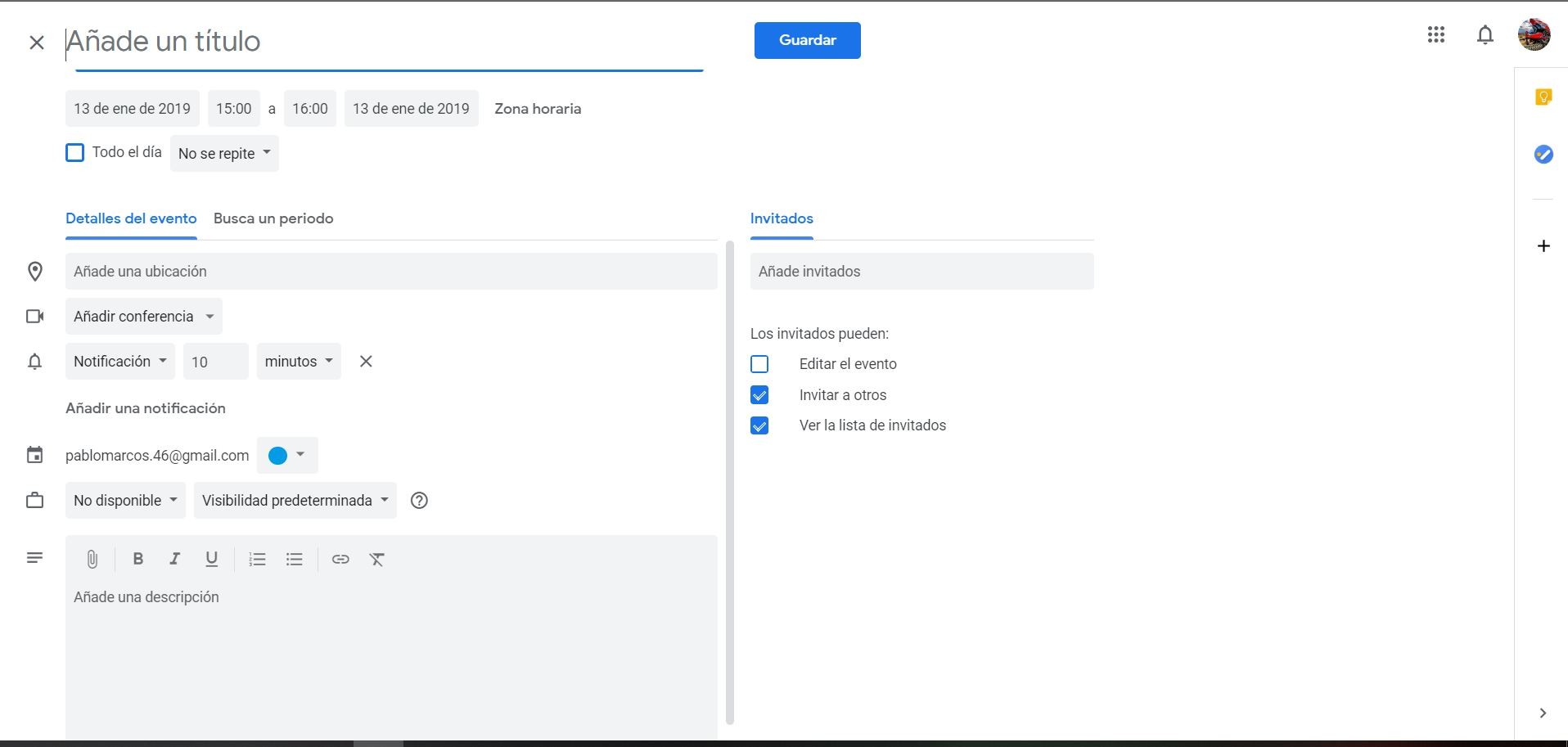
Confirmar o rechazar la asistencia a los eventos. Comprobación de asistencia de los invitados.

Creación de recordatorios.

Compartir y ver calendarios de otros usuarios. Personalizar el calendario

Posibilidad de ver el calendario por días, semanas, meses o años.





**Tabla comparativa de las aplicaciones estudiadas**

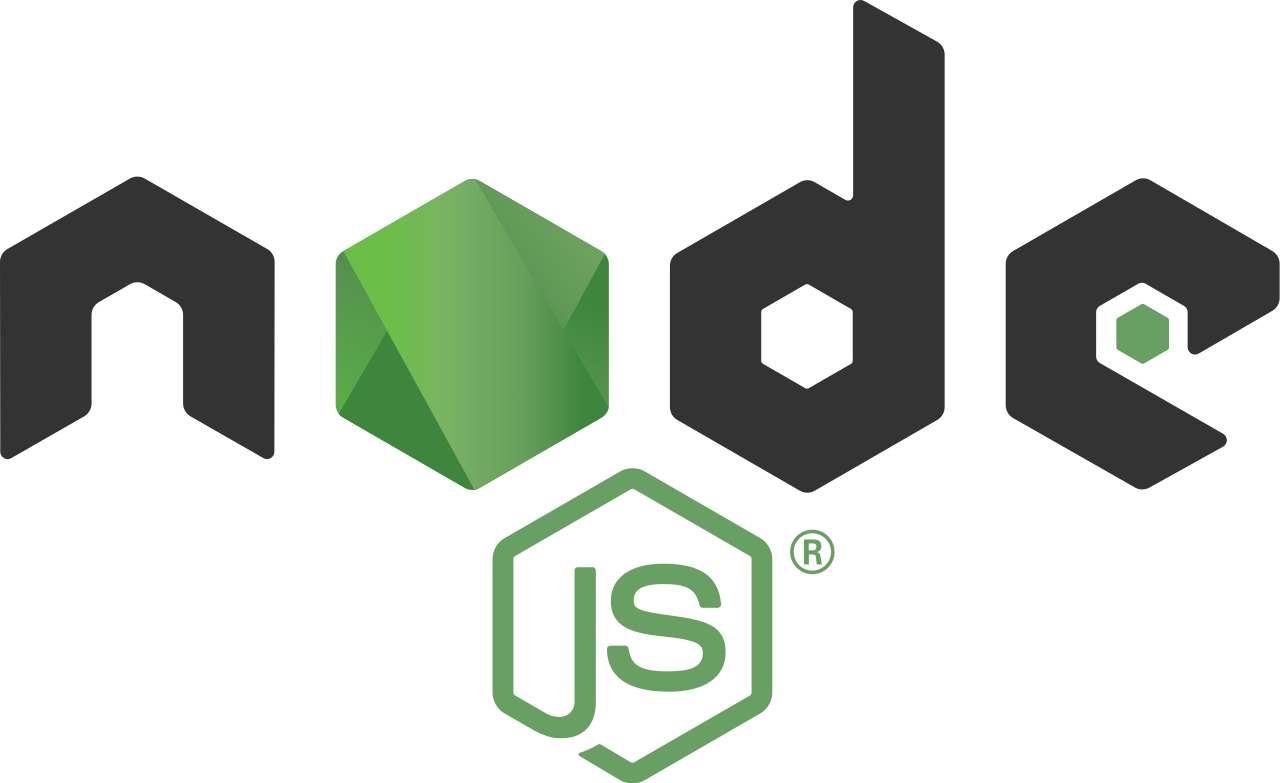
En este apartado se procederá a realizar una comparativa entre las aplicaciones, analizando tanto el tipo de licencia que se necesita para poder utilizarlas como la posibilidad o no de realizar ciertas funcionalidades que se necesitan en la aplicación que se va a desarrollar en el presente [TFG](#_bookmark7).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **GoToMeeting** | **Doodle** | **Joan** | **Google Calendar** |
| **Tipo de licencia** | Suscripción por mes | Suscripción por año | Pago único | Gratuita (uso personal) |
| **Creación de eventos** | Sí | Sí | Sí | Sí |
| **Reserva de salas** | No | No | Sí | Sí |
| **Adjuntar documentos** | Sí | No | Sí | Sí |
| **Añadir asistentes a las reservas** | Sí | Sí | Sí | Sí |
| **Gestión de conflictos** | No | No | No | No |
| **Gestión de reservas** | Sí | Sí | Sí | Sí |
| **Notificaciones por correo** | Sí | Sí | Sí | Sí |
| **Comentarios en reservas** | Sí | No | No | No |

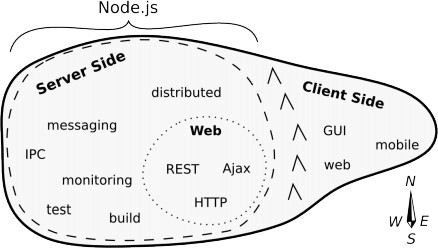
## Tecnologías

En este apartado se desglosarán todas las tecnologías que se utilizarán para desarrollar la aplicación objetivo de este

### Node.js

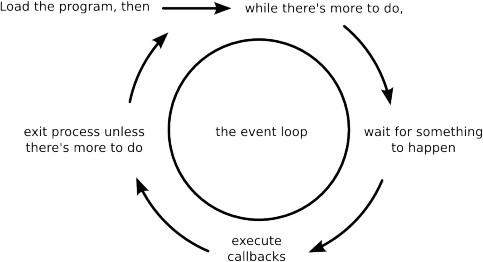
Node.js es un entorno de ejecución para JavaScript construido con el motor V8 de Chrome.

Para explicar dónde ubicamos Node.js en una arquitectura web, Jim R.Wilson expone un curioso ejemplo que agrupa todos los programas posibles en una isla.



**Funcionamiento de una aplicación en Node.js**

La principal característica del funcionamiento de Node.js es que combina JavaScript con un bucle de eventos para enviar rápidamente operaciones, y así poder ejecutar distintos blo- ques de código de forma paralela.



Siempre que haya tareas pendientes por ejecutarse, el bucle de eventos seguirá girando, como se ve en la Figura [3.10.](#_bookmark21) Si se dispara un evento, Node invocará todos los *callbacks* que se encuentren a la escucha de dicho evento.

Un *callback* es un bloque de código que se ejecutará como resultado de un evento, y devolverá a quien lo invocó el propio resultado o un mensaje del error que se haya podido producir durante el proceso.

#### Características principales de Node.js

Todo el código se ejecutará en el servidor, por lo que la programación no tendrá res- tricciones.

La programación es asíncrona. De esta manera se evitarán bloqueos entre instrucciones que no dependan entre ellas.

Permite modularizar el código en todos los ficheros que sean necesarios, haciendo que la comprensión de la estructura de un proyecto llegue a ser sencilla.

### Express.js

[Express](#_bookmark67) es un framework desarrollado para Node.js, que fue diseñado para construir aplicaciones web y APIs.



Este framework se instala junto a Node y permite generar un esqueleto funcional de una aplicación web simplemente ejecutanto el comando “express [nombre aplicación]”.

En el desarrollo de la aplicación objetivo de este [proyecto](#_bookmark7), se utilizará Express.js para gene- rar el esqueleto de la parte de la aplicación que se ejecutará en el servidor, recibiendo y procesando las peticiones de los clientes.

## Medios utilizados

En este apartado se detallan los medios utilizados para el desarrollo de la aplicación obje tivo de este [proyecto](#_bookmark7) , ya sean equipos o software.

### Medios hardware

Se han utilizado dos equipos para el desarrollo del presente [proyecto](#_bookmark7):

Como equipo de trabajo para desarrollar la aplicación se ha utilizado un equipo portátil de pr con las siguientes especificaciones:

**Procesador:** Intel(R) Core i5 5200U 2.2GHz

**RAM:** 8GB

**Almacenamiento:** 500GB HDD

**Sistema Operativo:** Windows 10 Pro x64

### Medios software

Las herramientas software utilizadas para el desarrollo del [proyecto](#_bookmark7), enumeradas según su función, son las siguientes:

Sistemas Operativos

**Windows 10 Pro** [[29],](#_bookmark68) como sistema operativo del equipo de trabajo utilizado para desarrollar la aplicación.

Entornos de desarrollo

**Visual Studio Code** utilizado tanto para tareas de codificación como de depura- ción de la aplicación.

Herramientas de gestión y planificación

**Visual Paradigm** [,](#_bookmark68) utilizado para la creación de modelos en la fase de diseño de la aplicación.

**Git - GitLab** [],](#_bookmark67) utilizado para almacenar el repositorio del proyecto.

**Microsoft Office 365** [,](#_bookmark68) utilizado para la elaboración de todo lo relacionado con la documentación de la aplicación.

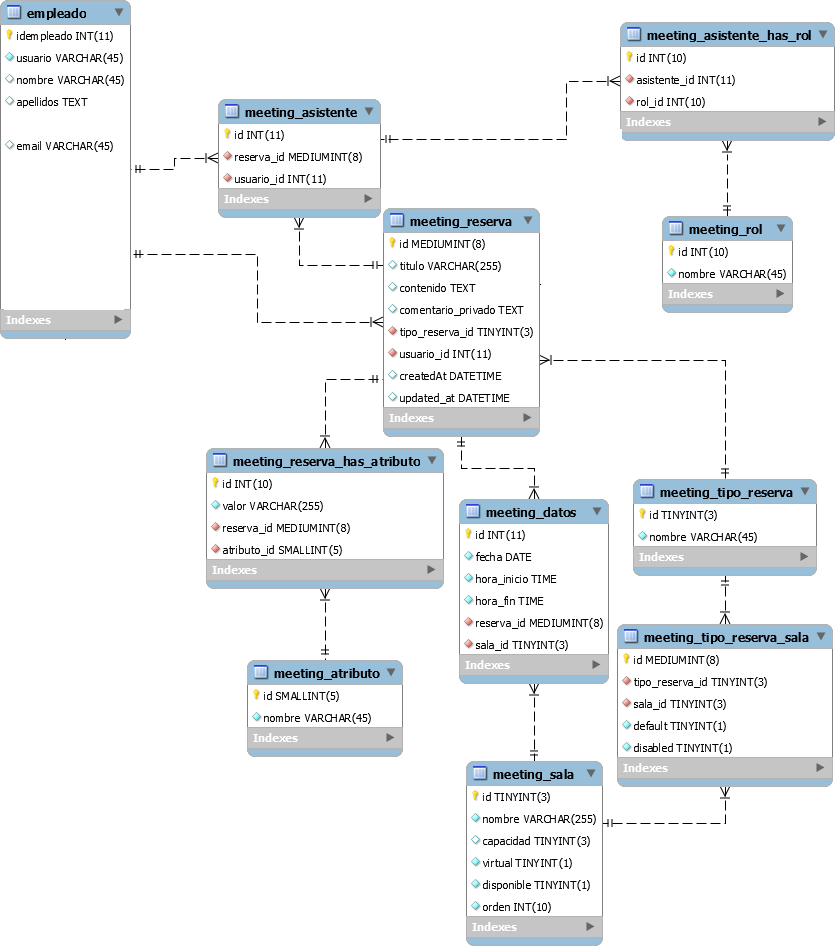
**MobaXTERM** [[19],](#_bookmark68) herramienta utilizada para acceder de forma gráfica y remota al servidor de pruebas de [ZENNIO](#_bookmark7).

**MySQL Workbench** [[21],](#_bookmark68) herramienta gestora de bases de datos, utilizada para ges- tionar de forma eficiente tanto la estructura como los datos de la base de datos de la aplicación.

**XAMPP** [[30],](#_bookmark68) paquete de servicios para el desarrollo de aplicaciones web, utilizado para mantener en el equipo de trabajo un servidor local de MySQL, donde se encuentra almacenada la base de datos utilizada durante el desarrollo.

## Resultados

La Figura muestra el diagrama Entidad-Relación de la base de datos obtenido, y que se implantará en la aplicación final. Cabe aclarar que la tabla *empleado* contiene campos a los que la aplicación objetivo de este [proyecto](#_bookmark7) no va a acceder, y además almacenan información delicada de los empleados. Por lo tanto, se ha decidido censurar esos campos.



## Diseño de la interfaz

El diseño de la interfaz será la última decisión a tomar antes de comenzar con la imple- mentación.

Diseñar una buena interfaz es una parte fundamental, ya que la interfaz será la parte que pondrá en contacto a los usuarios finales con la aplicación, y su *usabilidad* determinará el éxito o fracaso de la decisión tomada en este punto.

La **usabilidad** es una medida que determina qué nivel de calidad que experimentan los usuarios con el uso de un producto o sistema. Esta medida no se puede saber con exactitud mientras el usuario no disponga de un producto tangible que pueda probar, pero se puede conseguir una predicción de la usabilidad que puede llegar a tener un sistema mediante la creación de bocetos.

En [ZENNIO](#_bookmark7), al igual que en la fase de requisitos, para decidir cómo será la interfaz de la aplicación se establece una reunión convocando a las mismas personas

En dicha reunión se presentan varios bocetos como candidatos de cada una de las vistas que compondrán la aplicación, se explica qué ventajas e inconvenientes podrían tener, y al final de la reunión se decide cuáles serán los bocetos que se utilizarán como guía en la implementación de la interfaz final.

Los bocetos presentados y divididos por vistas, se presentan en los siguientes apartados.

## Vista de inicio

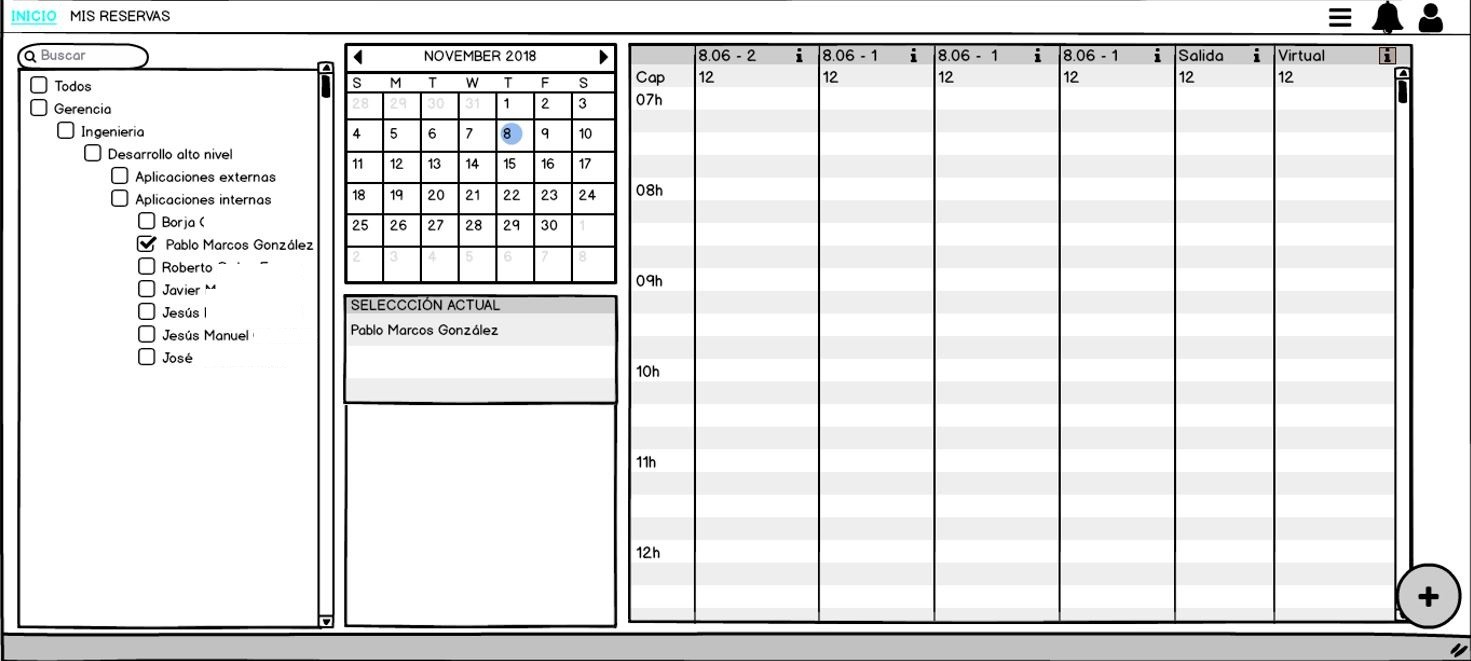
La vista de inicio será una vista de consulta de información, en la que se podrá ver el estado de las salas para una fecha determinada, y seleccionando empleados de una estructura en árbol se podrá visualizar en la tabla de salas si están ocupados en alguna franja horaria.

Se presenta una disposición en forma de columnas, distinguiéndose tres partes principales

La parte izquierda muestra la estructura en forma de árbol con todos los empleados y departamentos de [ZENNIO](#_bookmark7).

La parte central muestra, en la parte superior, el calendario seleccionable, y en la parte inferior, la lista de empleados seleccionados.

La parte derecha muestra la tabla de selección de sala y franja horaria, pero de forma invertida respecto a como se muestra en el *Candidato 1* y *Candidato 2*. Las horas se ven en las filas, y las salas se ven en las columnas.



## Implementación

En esta etapa se procede a realizar el desarrollo de la aplicación objetivo de este [TFG](#_bookmark7).

Debido a la aplicación de la metodología [MPIU+A](#_bookmark7), la implementación será estructurada por funcionalidades, de forma que después de cada fase exista un producto mínimo que pueda ser probado por los usuarios finales para así ver si es viable continuar o es necesario pulir dicha funcionalidad, pero antes de implementar las funcionalidades hay que preparar los entornos tanto del cliente como del servidor.

Para el desarrollo de la aplicación se utilizará el Entorno de Desarrollo Integrado ([IDE](#_bookmark7)) Visual Studio Code [,](#_bookmark68) y para la gestión de la [BD](#_bookmark7), MySQL Workbench [.](#_bookmark68)

En los siguientes apartados se explicará el procedimiento llevado a cabo para conseguir la aplicación final.

**Estructuración del *backend***

El *backend* es la parte de la aplicación cuyas funcionalidades se ejecutan en el servidor.

Esta parte tiene implementados los siguientes módulos:

[**API**](#_bookmark7) **de comunicaciones websockets**: Funciones relacionadas con la comunicación mediante websockets.

[**API**](#_bookmark7) **de comunicaciones HTTP.**: Funciones relacionadas con la comunicación me- diante el protocolo [HTTP](#_bookmark7).

**Componentes:** Elementos principales e independientes de la lógica de negocio. En cada componente se distinguen dos partes:

**Business Logic Layer (**[**BLL**](#_bookmark7)**), o *Capa de Lógica de Negocio*:** es la parte que contiene las funciones que procesan la información de la aplicación.

**Data Access Layer (**[**DAL**](#_bookmark7)**), o *Capa de Acceso a Datos*:** es la parte que se encarga de comunicar la aplicación con la [BD](#_bookmark7).

**Modelos de datos:** ficheros que representan las tablas de la [BD](#_bookmark7) en forma de modelos, con sus atributos, relaciones y validaciones.

**Utils:** funciones accesibles por cualquier parte del *backend* con finalidades de tipo genérico (por ejemplo, cambiar el formato de una fecha).

Una vez definida la estructura general del *backend*, se procede a crear la estructura de carpetas y ficheros que lo compongan, tomando como base lo enumerado anteriormente.

## Conclusiones

El objetivo principal descrito en este [proyecto](#_bookmark7) era desarrollar una aplicación web que ayudase a en las tareas relacionadas con la gestión de reserva de salas y tiempo de trabajo, además de almacenar un histórico tanto de las reservas que se harán en el día a día como de las reuniones ya finalizadas.

El objetivo principal de este [proyecto](#_bookmark7) ha sido alcanzado, y por tanto, también han sido logrados los objetivos parciales que lo componen. Todo ello ha sido documentado en este documento.

A continuación, se justifica la consecución de cada uno de los objetivos:

**Objetivo ¿Conseguido?**

Desarrollar una aplicación web que ayude en las tareas relacionadas con la gestión de reserva de salas y tiempo de trabajo, además de almacenar un histórico tanto de las reservas que se harán en el día a día como de las reuniones ya finalizadas.



## Mejoras futuras

Durante el desarrollo de este [proyecto](#_bookmark7), han surgido nuevas ideas que en un futuro serán materializadas como nuevas funcionalidades de la aplicación que se ha desarrollado.

**Sistema de notificaciones:** En futuras versiones, se procederá a implementar un sistema de notificaciones por correo, para así mantener informados a los usuarios de cualquier cambio que puedan afectar a las reservas de las que son participantes.

**Ayudas en la selección de salas:** Se implementarán mecanismos de ayuda a la hora de seleccionar salas, habilitando y deshabilitando franjas horarias según la disponibilidad de los usuarios seleccionados como participantes.

**Implementación de protocolos** [**HTTPS**](#_bookmark7) **y WebSockets Seguro (**[**WSS**](#_bookmark7)**):** Con la finali- dad de hacer toda comunicación segura, se actualizarán las [API](#_bookmark7)s implementadas para que utilicen protocolos seguros.

**Plantillas para reuniones predefinidas:** El usuario podrá crear una plantilla basán- dose en alguna reserva previa. Esta plantilla almacenaría los participantes y sus roles, así como todos los datos de la reunión (título, contenido y tipo de reunión). Estas plan- tillas podrían cargarse fácilmente desde un selector situado en algún punto de la vista de creación de reserva o la vista de inicio de la aplicación.

# ANEXOS

## Manual de usuario de la aplicación

### Vista de inicio

Cuando acceda a la aplicación, el usuario se encontrará en una vista como la mostrada en la Figura [A.1.](#_bookmark69) En ella, podrá consultar entre los empleados de [ZENNIO](#_bookmark7) (parte superior izquierda), si se encuentran ocupados en ciertas fechas.

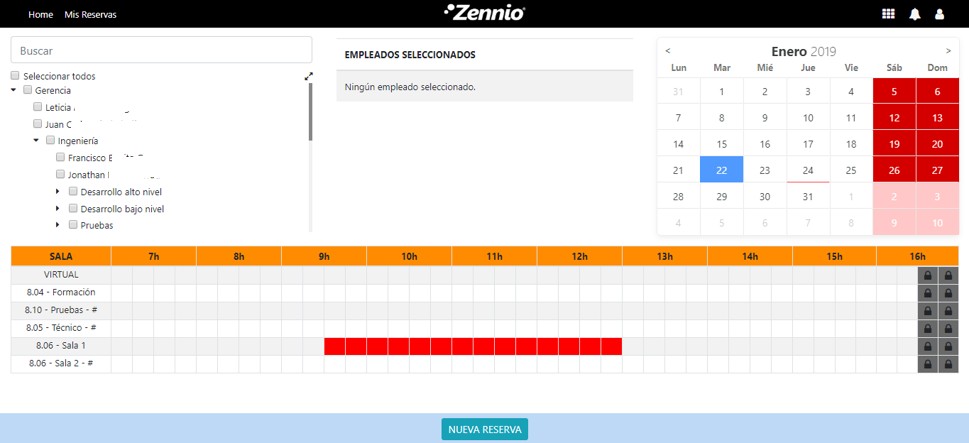


Figura A.1: Vista de inicio

Para realizar una consulta, el usuario seleccionará de la estructura en árbol el empleado que requiera, y en el calendario (parte superior derecha) se visualizarán de color naranja los días en los que el empleado a consultar esté ocupado en algún momento del día.

En caso de que el usuario quiera ver dentro de uno de los días marcados en naranja qué franja horaria y en qué sala está ocupado el empleado a consultar, sólo tendrá que pulsar sobre el día a consultar en el calendario, y la tabla de salas y franja horaria (parte inferior) cambiará, visualizándose con un icono la franja horaria en la que el usuario está ocupado (véase Figura [A.2).](#_bookmark70)

Para consultar al mismo tiempo varios empleados, la operación es la misma solo que el usuario tendrá que seleccionar antes a todos los empleados a consultar en la estructura en árbol.

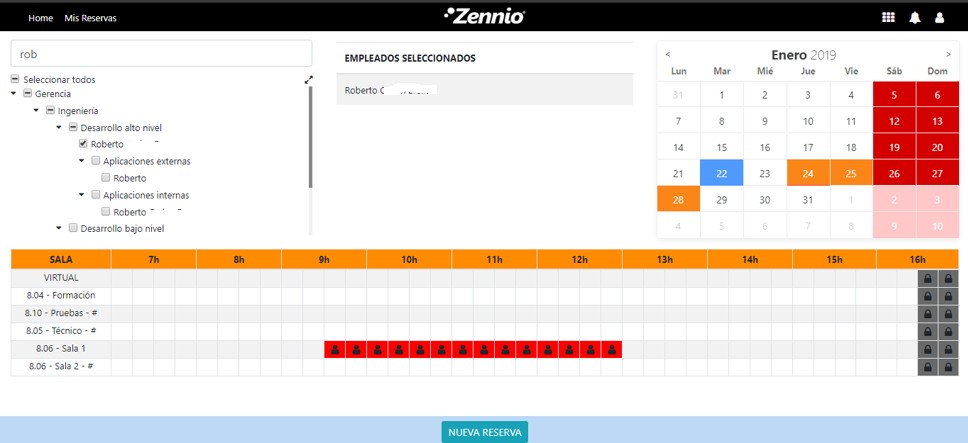


Figura A.2: Mostrando conflictos en un día

Si el usuario coloca el cursor encima del icono de conflicto, se desplegará una lista en la que se mostrarán los empleados que se encuentran ocupados en esa franja horaria (véase Figura [A.3).](#_bookmark70)

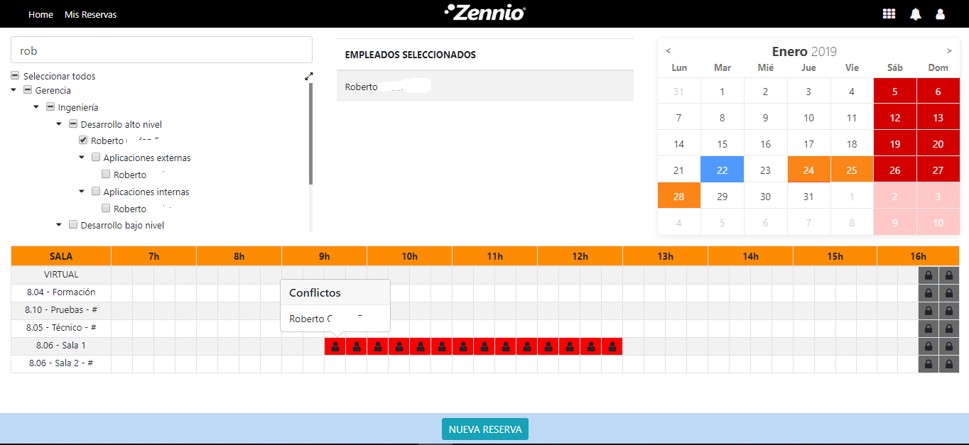


Figura A.3: Listado de empleados ocupados

Como muestra la Figura [A.4,](#_bookmark71) la tabla de salas es seleccionable, permitiendo marcar una franja horaria y sala deseadas. Esto permite que la selección se quede almacenada, y en caso de pulsar el botón de ‘NUEVA RESERVA’, la selección realizada tanto de horas como de empleados se introducirá en la vista de ‘creación de reserva’ (véase Figura [A.5).](#_bookmark71)

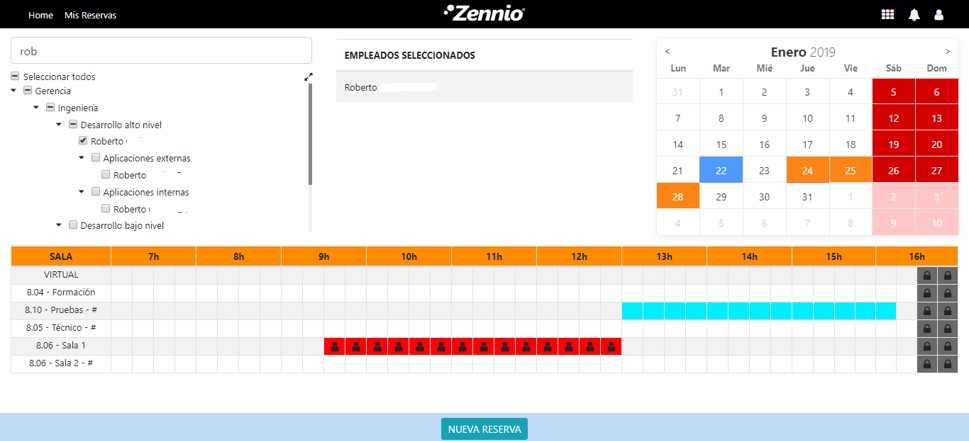


Figura A.4: Selección en tabla de salas

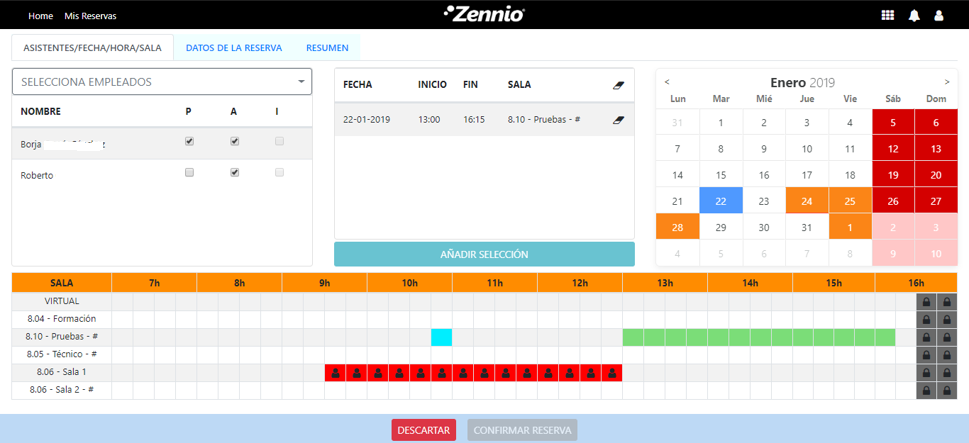


Figura A.5: Datos guardados en vista de nueva reserva

## Alta de reserva

La creación de reservas es la funcionalidad principal de esta aplicación, y la que más pasos necesita para ser completada. Por ello, en este apartado se detallarán lo máximo posible los pasos a seguir para poder completar un alta de reserva, y qué errores se puede encontrar el usuario durante el procedimiento.

Cuando el usuario acceda a la vista de creación de reserva, verá lo mostrado en la Figura

[A.6.](#_bookmark72) En la lista de participantes (parte superior izquierda), el mismo usuario que esté reali- zando la operación de creación aparecerá como asistente y propietario de la reserva, y si se diese el caso de tener un responsable a su cargo, éste aparecería como informado en la lista en un tono más difuminado. Esto quiere decir, que dicho informado podrá ser desmarcado y podría no tener ningún rol en la reserva, descartándose así de la lista de participantes.

El calendario (parte superior derecha) se mostrará por defecto en el mes y día actuales, de forma que el usuario verá en la tabla de salas (parte inferior) la disposición de franjas horarias libres para el mismo día en el que realiza la reserva.

Por defecto, el botón de ‘CONFIRMAR RESERVA’ se encontrará deshabilitado, lo que querrá decir que se deberán cumplir ciertos requisitos para que una reserva sea válida. Los requisitos a cumplir son los siguientes:

Participantes:

Al menos uno de los participantes seleccionados deberá ser propietario.

Todos los participantes seleccionados por el usuario deberán tener al menos un rol asignado.

Fechas/Horas/Salas:

Deberá haber, como mínimo, una selección almacenada.

Ninguna selección realizada deberá estar en conflicto con otra selección realizada por otro usuario en el momento de crear la reserva (casillas de color naranja en tabla de salas y horas).

Datos de la reserva:

o El usuario deberá cumplimentar para una reserva el ‘Título’ y el ‘Tipo de re- serva’ (ambos marcados con un ‘\*’).

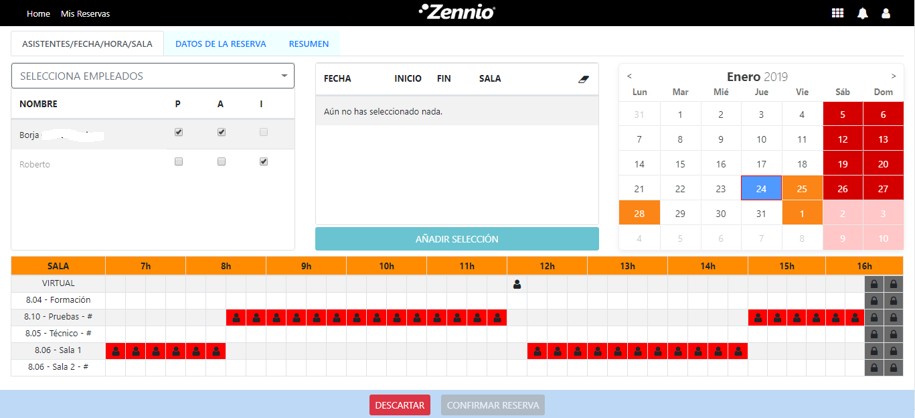


Figura A.6: Nueva reserva vacía

## Selección de participantes, fechas, horas y salas

El primer paso a realizar por el usuario en la creación de una reserva será seleccionar los participantes de la reserva. Para ello, desplegará la estructura en forma de árbol que contiene a todos los empleados de [ZENNIO](#_bookmark7) (véase Figura [A.7),](#_bookmark73) pulsando en el desplegable ubicado en la parte superior de la lista de participantes.

Una vez desplegada la estructura en árbol, seleccionará el empleado o departamentos que quiera que sean participantes. Al seleccionarlos, se agregarán a la lista de participantes como ‘Asistentes’.



Figura A.7: Selección de participantes

La lista de participantes (Figura [A.8)](#_bookmark73) permite intercambiar los roles de cada empleado seleccionado, pudiendo ser éstos ‘Propietario’, ‘Asistente’ o ‘Informado’.

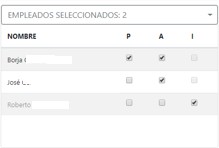


Figura A.8: Listado de participantes

Seleccionados los participantes y asignados sus roles, el usuario procederá a realizar la selección de días, horas y salas en las que se realizará la reunión.

Primero, seleccionará el día deseado (Figura [A.9).](#_bookmark74) Esto refrescará la tabla de salas (Figura [A.10)](#_bookmark74) mostrando las franjas horarias ocupadas y libres, y mediante iconos, las franjas en las que alguno de los participantes seleccionados con rol ‘Asistente’ esté ocupado.



Figura A.9: Selección del día

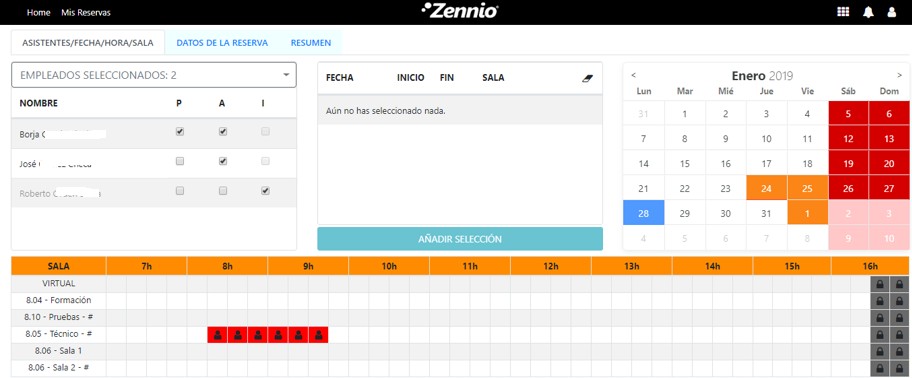


Figura A.10: Tabla de salas en vista de creación

Para realizar una selección de franja horaria, el usuario pulsará sobre una franja concreta, y sin soltar el botón arrastrará el cursor a lo largo de la franja a reservar (Figura [A.11).](#_bookmark74) Esto creará una selección en la tabla de salas y horas.

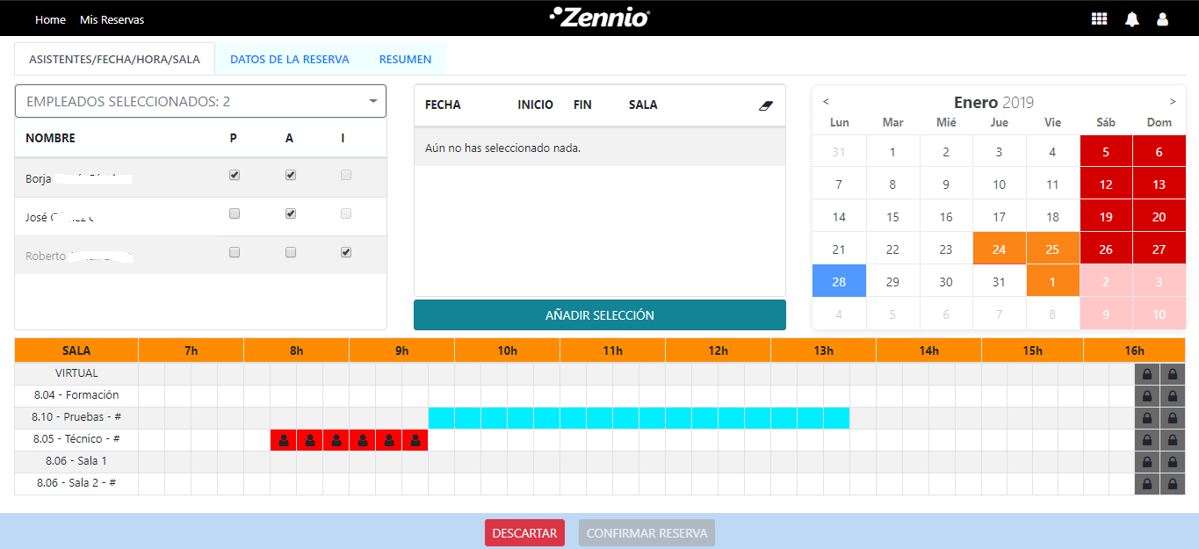


Figura A.11: Selección de hora y sala

Una vez hecha la selección, el botón de ‘AÑADIR SELECCIÓN’ se habilitará, permitien- do al usuario almacenar dicha selección en la lista ubicada justo encima del botón (Figura [A.12).](#_bookmark75)



Figura A.12: Botón de añadir selección habilitado

Si el usuario pulsa sobre el botón de ‘AÑADIR SELECCIÓN’, la selección quedará al- macenada en la lista (Figura [A.13),](#_bookmark75) permitiendo cambiar la fecha del calendario a la fecha indicada en la selección en cualquier momento si el usuario pulsa en la fila del registro.



Figura A.13: Fecha almacenada

Los iconos que se encuentran a la derecha tanto de la cabecera como de cada una de las filas de la lista, sirven para eliminar los registros de forma individual (iconos en las filas), o bien para eliminar todos los registros (icono en la cabecera). Las operaciones de borrado tendrán mensajes de confirmación.

La selección que se haya realizado se quedará marcada de color verde en la tabla de salas (Figura [A.14).](#_bookmark76)

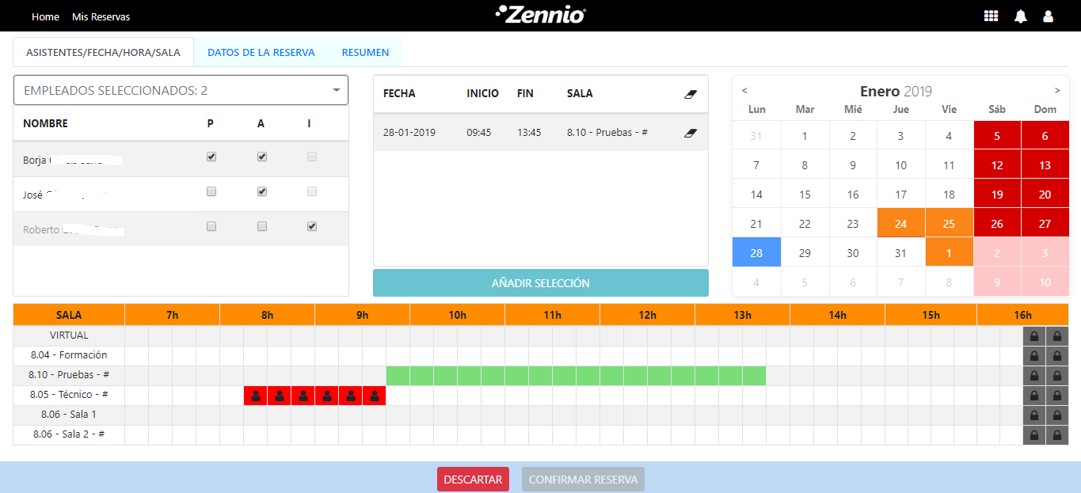


Figura A.14: Selección guardada en tabla de salas

A su vez, los días en los que se haya hecho alguna selección, se mostrarán de color verde en el calendario (Figura [A.15).](#_bookmark76)

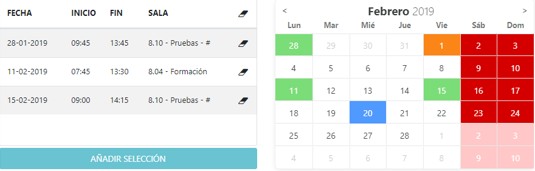


Figura A.15: Fechas seleccionadas en calendario

## Formulario de datos de la reserva

El siguiente paso a realizar por el usuario, será rellenar los datos de la reserva. Para ello pulsará sobre la pestaña ‘DATOS DE LA RESERVA’. Se mostrará el formulario de datos de la reserva (Figura [A.16).](#_bookmark77)

Los datos que deberá cumplimentar de forma obligatoria serán generalmente ‘TÍTULO’ y ‘TIPO DE RESERVA’. El resto de ellos serán opcionales.

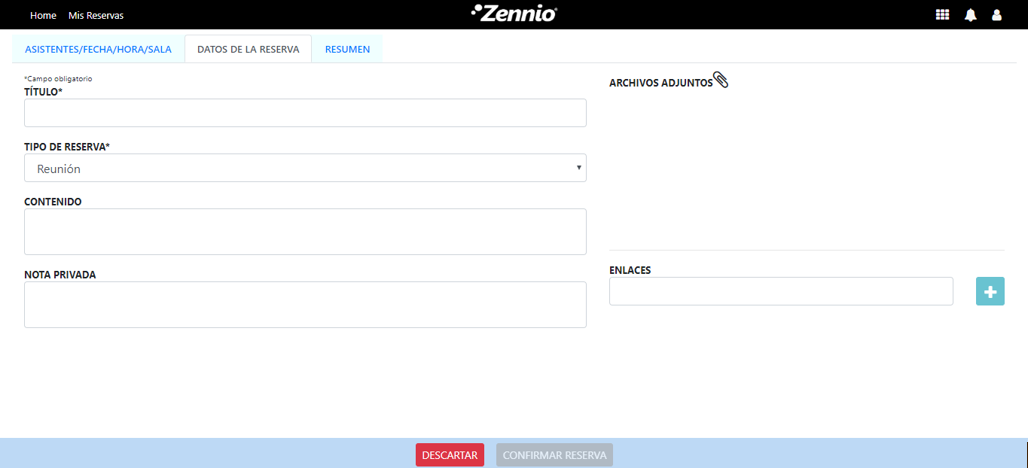


Figura A.16: Formulario de datos de reserva

El campo ‘TIPO DE RESERVA’ es un desplegable, y su información dependerá de las salas que se hayan seleccionado en el paso anterior (Selección de participantes, fechas, horas y salas). Por defecto, se seleccionará el tipo de reserva contenido por defecto en la primera sala que el usuario eligió.

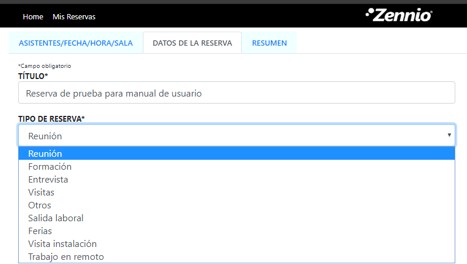


Figura A.17: Título y tipo de reserva

Cumplimentados los datos obligatorios (Figura [A.18),](#_bookmark78) y habiendo llevado a cabo de forma correcta el paso anterior, el botón de ‘CONFIRMAR RESERVA’ se habilitará permitiendo completar la reserva (Figura [A.19).](#_bookmark78)

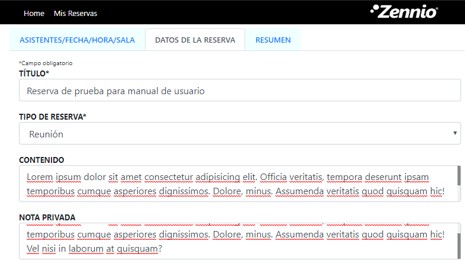


Figura A.18: Datos de reserva cumplimentados



Figura A.19: Botón de confirmar habilitado

Si el usuario selecciona en el paso anterior una sala virtual (VIRTUAL o SALIDAS), en el formulario aparecerá un campo extra llamado ‘LOCALIZACIÓN’ (Figura [A.20),](#_bookmark78) que será obligatorio en adición a los dos campos descritos anteriormente.

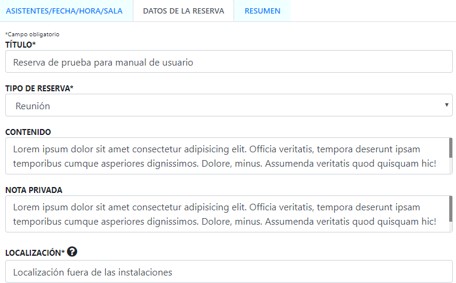


Figura A.20: Campo de localización

**El siguiente paso es opcional:** Si el usuario ha llegado a este punto, tendrá los datos obligatorios cumplimentados y podrá realizar la reserva, pero si lo desea, podrá además adjuntar archivos y enlaces (Figura [A.21).](#_bookmark79)

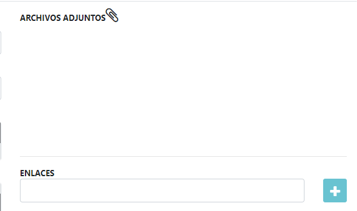


Figura A.21: Adjuntos y enlaces

Para adjuntar uno o varios archivos, el usuario deberá pulsar sobre el icono de ‘clip’ situa- do a la derecha del título de ‘ARCHIVOS ADJUNTOS’. Aparecerá una ventana como la que muestra la Figura [A.22.](#_bookmark79) En ella, seleccionará los archivos a adjuntar y pulsará en el botón de “Abrir”.

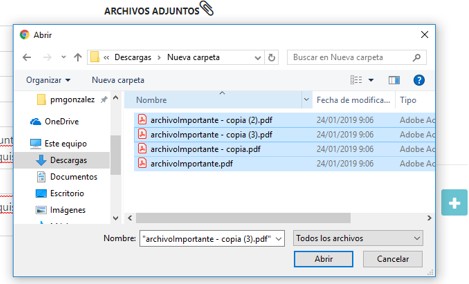


Figura A.22: Ventana para adjuntar archivos

Los archivos seleccionados aparecerán en la lista que muestra la Figura [A.23.](#_bookmark80)

Los archivos adjuntos se podrán eliminar, pulsando sobre los iconos de ‘borrador’ situados a la derecha de cada fila de la lista. Las operaciones de borrado mostrarán previamente un mensaje de confirmación.



Figura A.23: Lista de adjuntos

Para agregar enlaces, el usuario rellenará el campo de texto con el enlace a añadir (Figura [A.24),](#_bookmark80) y a continuación pulsará sobre el botón que contiene el ‘+’. Esto añadirá el enlace a una lista situada justo debajo del campo de texto (Figura [A.25).](#_bookmark80)



Figura A.24: Insertar enlace

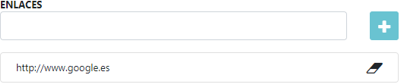


Figura A.25: Enlace insertado

Al igual que con los archivos adjuntos, se habilita al usuario una acción de borrado pulsan- do sobre el icono de ‘borrador’, ubicado a la derecha de cada enlace de la lista de añadidos. Se mostrará en pantalla un mensaje de confirmación, al igual que en la operación de borrado de adjuntos.

**Resumen de la reserva**

El último paso que puede realizar el usuario antes de confirmar la reserva, es revisar todos los datos cumplimentados para la misma. Pulsando sobre la pestaña ‘RESUMEN’, se mos- trará toda la información de la reserva cumplimentada en las dos pestañas anteriores (Figura [A.26).](#_bookmark81)

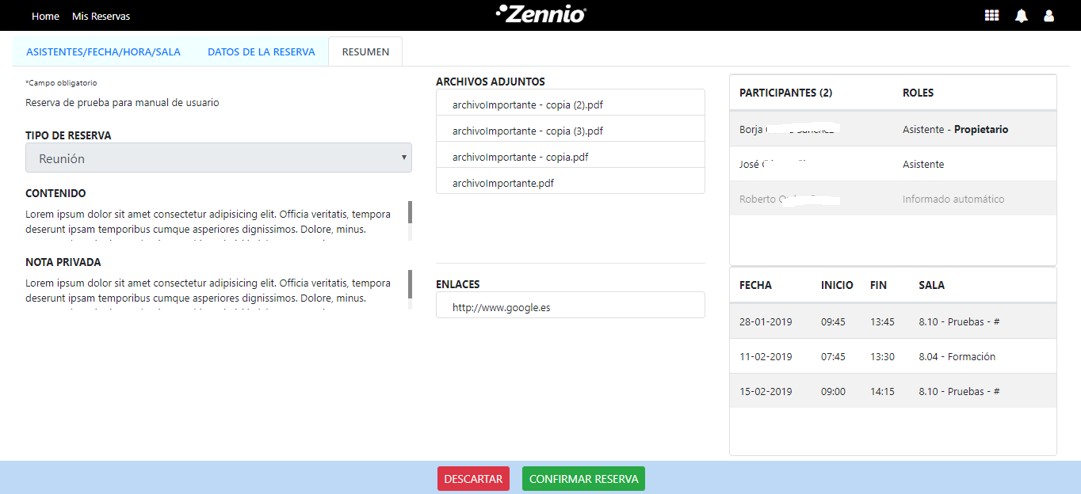


Figura A.26: Resumen de reserva

**Posibles problemas al crear reserva**

**Puede darse la siguiente situación:** El usuario ha completado los datos de la reserva, y se dispone a confirmarla. Pulsa sobre el botón de ‘CONFIRMAR RESERVA’, y la aplicación le devuelve el siguiente error:



Figura A.27: Error de conflicto

El error mostrado en la Figura [A.27](#_bookmark82) indica al usuario que otro usuario ha reservado una franja de las que el usuario que estaba realizando la reserva tenía almacenadas. Para com- probar el error, deberá navegar por las fechas seleccionadas hasta que encuentre una franja como la mostrada en la Figura [A.28.](#_bookmark82)

Para que la reserva sea válida, el usuario deberá cambiar esa franja horaria por otra que se encuentre libre.

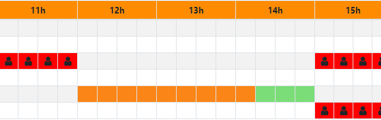


Figura A.28: Conflicto en franja horaria

**Puede darse la siguiente situación:** El usuario ha completado los datos de la reserva, y se dispone a confirmarla. Pulsa sobre el botón de ‘CONFIRMAR RESERVA’, y la aplicación le muestra el siguiente mensaje:



Figura A.29: Advertencia de conflicto

El mensaje mostrado en la Figura [A.29,](#_bookmark83) indica al usuario que uno de los participantes ya forma parte de otra reserva en una de las franjas horarias que han sido seleccionadas en la reserva actual, o bien se encuentra ausente en ese intervalo de tiempo. Esta advertencia podrá ser ignorada pulsando en ‘Aceptar’, pero en caso de querer comprobar el conflicto, la Figura

[A.30](#_bookmark83) muestra un ejemplo de un conflicto de este tipo. Se observa que, encima de la franja verde, hay una franja con iconos que indican que uno o varios empleados están ocupados en esa misma franja horaria, y, es posible que no puedan acudir a la convocatoria de ese día.

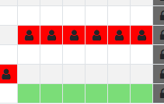


Figura A.30: Conflicto de empleados en franjas paralelas

**Confirmación de reserva**

Si el usuario pulsa en el botón de ‘CONFIRMAR RESERVA’, la aplicación comenzará a crear la reserva con toda la información cumplimentada.

Durante ese período, es posible que se muestre un icono de carga como el que muestra la Figura [A.31.](#_bookmark84)



Figura A.31: Icono de carga

Si todo el procedimiento de creación ha cursado sin error, se mostrará en la parte inferior derecha de la pantalla el siguiente mensaje:



Figura A.32: Mensaje de confirmación de reserva creada

Inmediatamente después de aparecer el mensaje mostrado en la Figura [A.32,](#_bookmark84) el usuario será redirigido a la vista de detalles de reserva (Figura [A.33).](#_bookmark84)

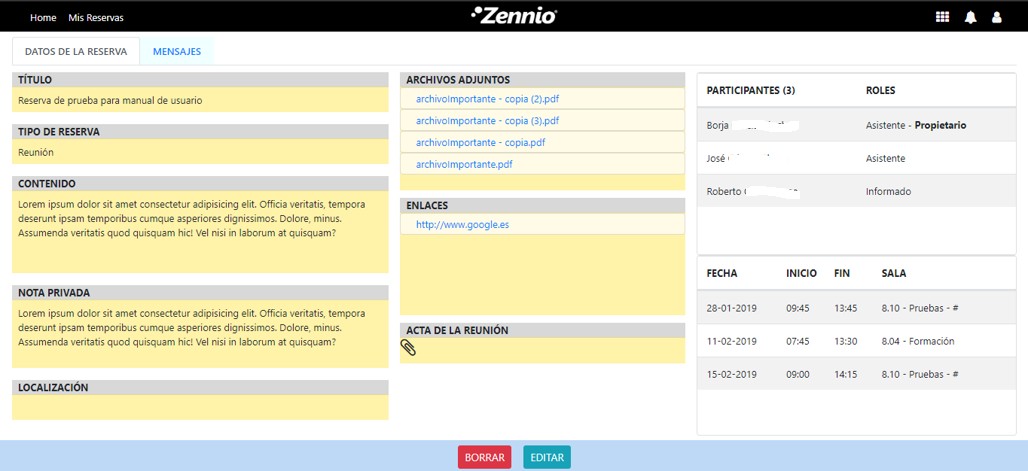


Figura A.33: Vista de detalles de reserva creada

**Gestión de Mis Reservas**

Otra funcionalidad que provee esta aplicación, es permitir al usuario llevar a cabo una gestión de las reservas en las que es participante. Para acceder a esta lista (Figura [A.34),](#_bookmark85) lo podrá hacer pulsando en el menú de la barra superior, en el enlace nombrado ‘Mis Reservas’.

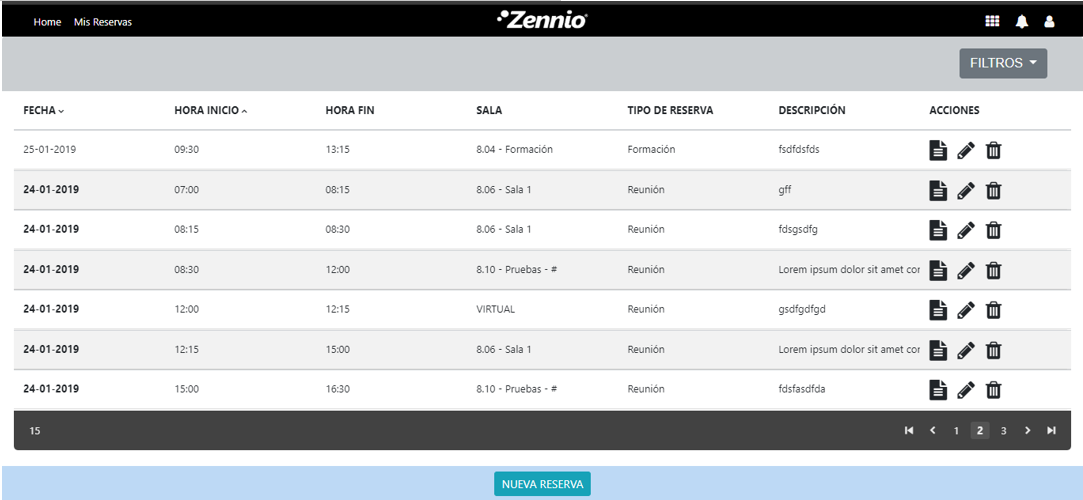
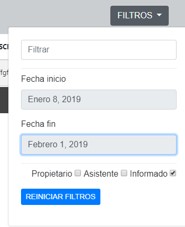


Figura A.34: Vista de ’mis reservas’

La tabla que se mostrará, desglosará los registros de datos de cada reserva (día o días en los que se convoca). En cada fila se visualizarán las fechas, horas, salas, tipo de reserva, descripción y acciones que el usuario puede realizar en cada registro. Las reservas que el usuario tenga programadas para el día en el que se encuentra, aparecerán con la fecha en negrita.

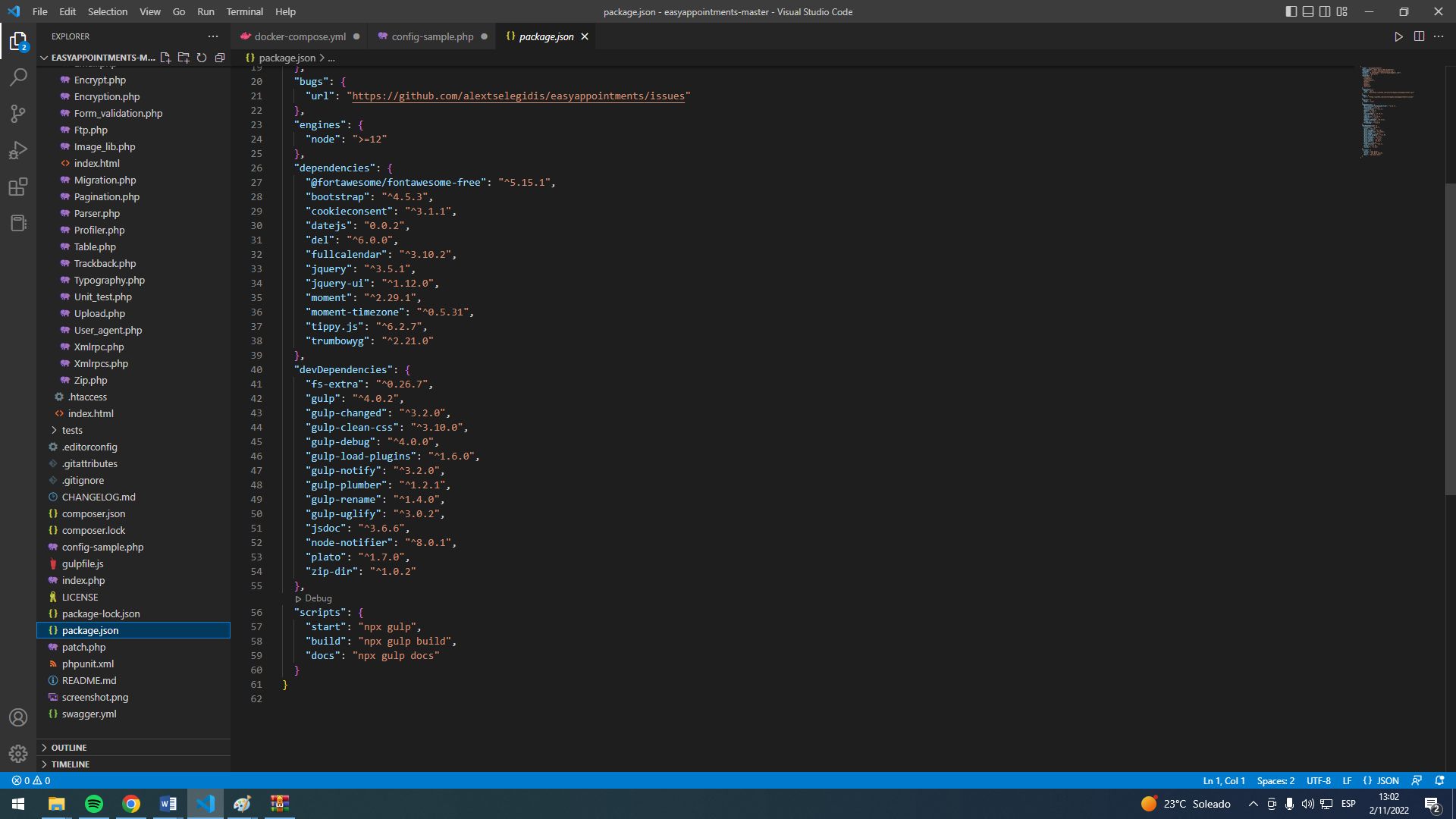
El usuario puede filtrar los datos, ya sea por texto (incluyendo todos los campos de cada registro), por rango de fechas o por roles dentro de cada reserva (Figura [A.35).](#_bookmark85) Si lo desea, el usuario puede reiniciar los filtros simplemente pulsando sobre el botón que se encuentra situado en la parte inferior del desplegable de filtros.



# PSEUDOCODIGO

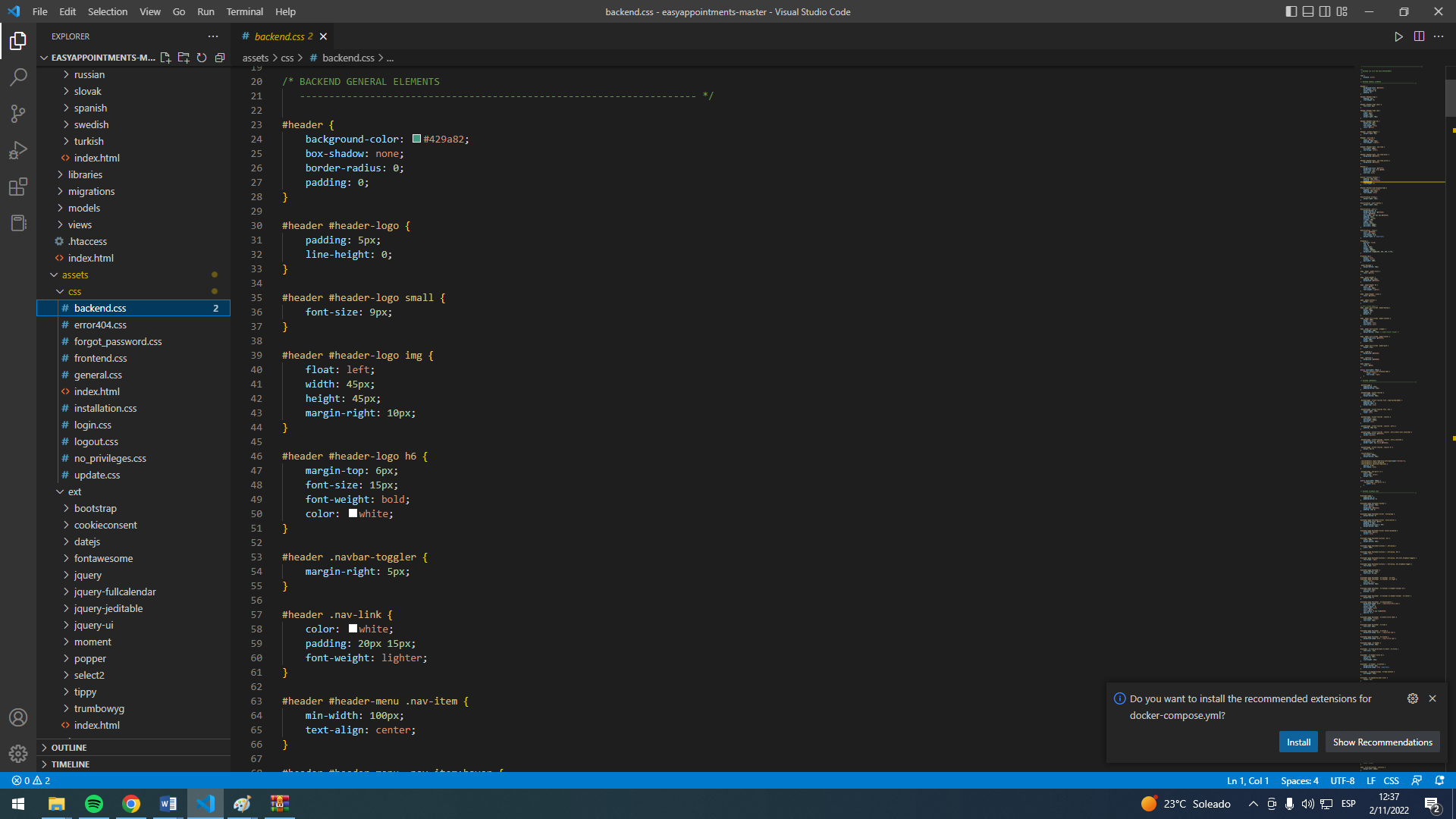
Dependencias

Dependencias necesarias para el proyecto:

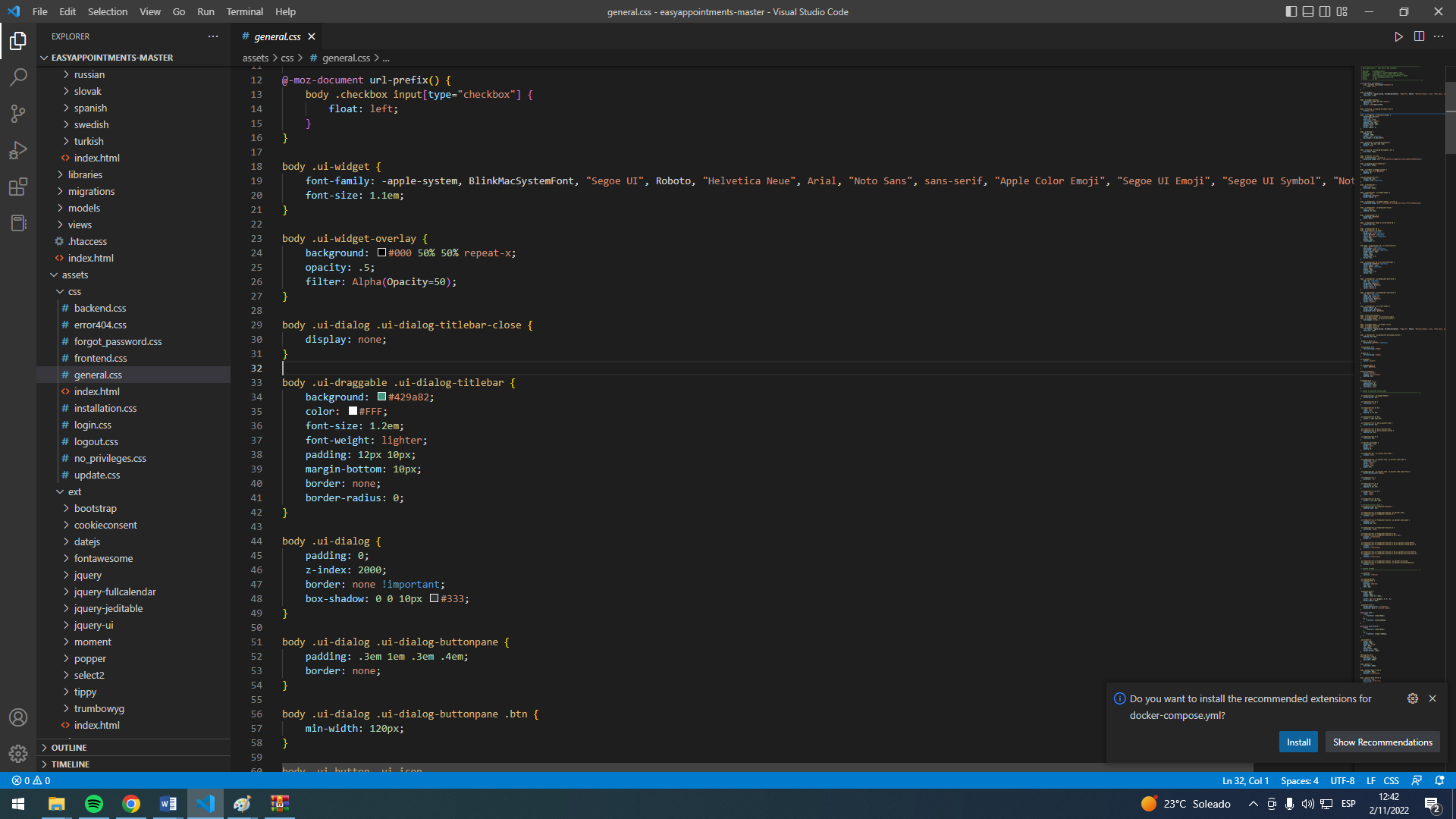


Estilos del Proyecto

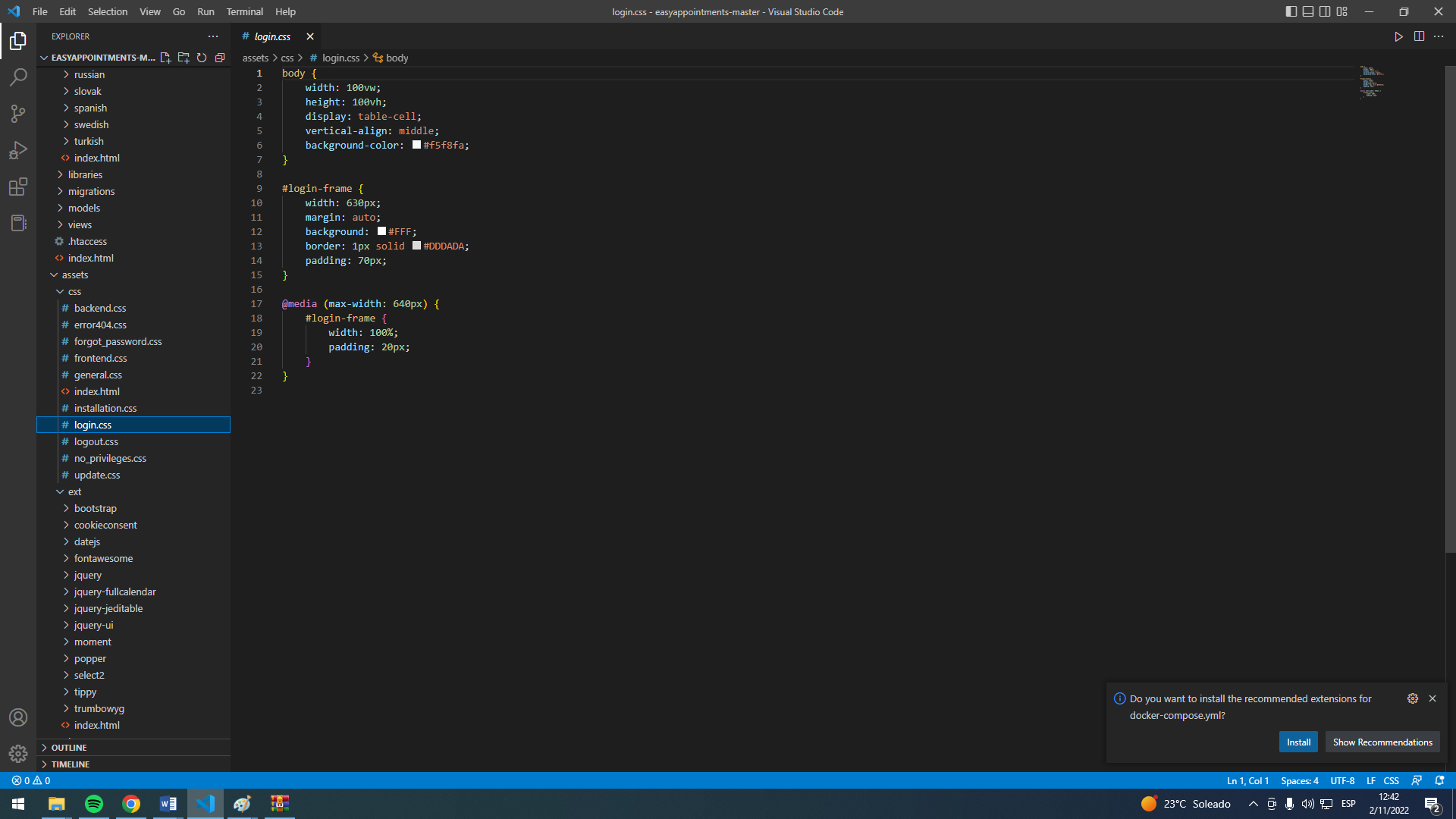
Empezamos el proyecto definiendo los estilos que se usaran para la interfaz del sistema



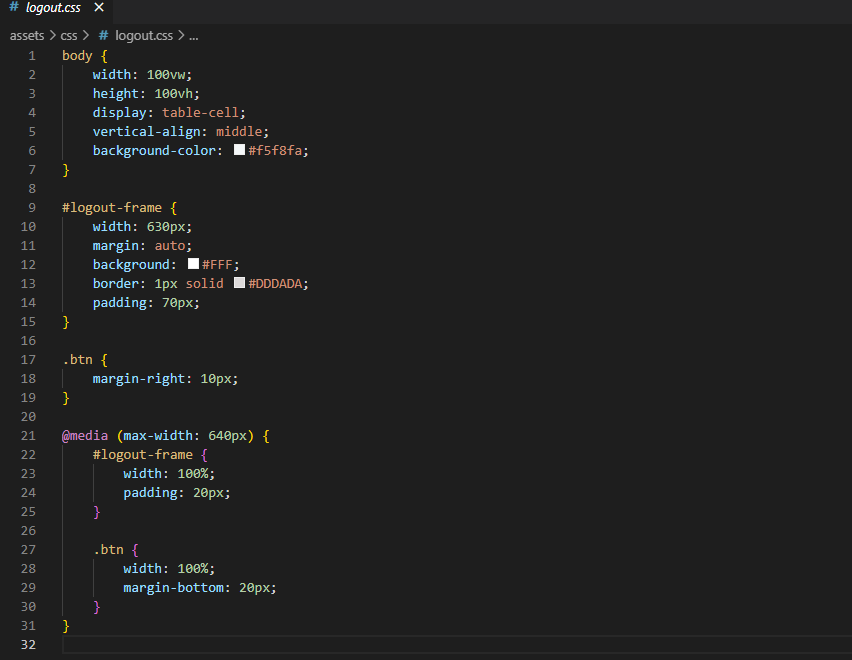
Para la vista en general:



Para el Login:

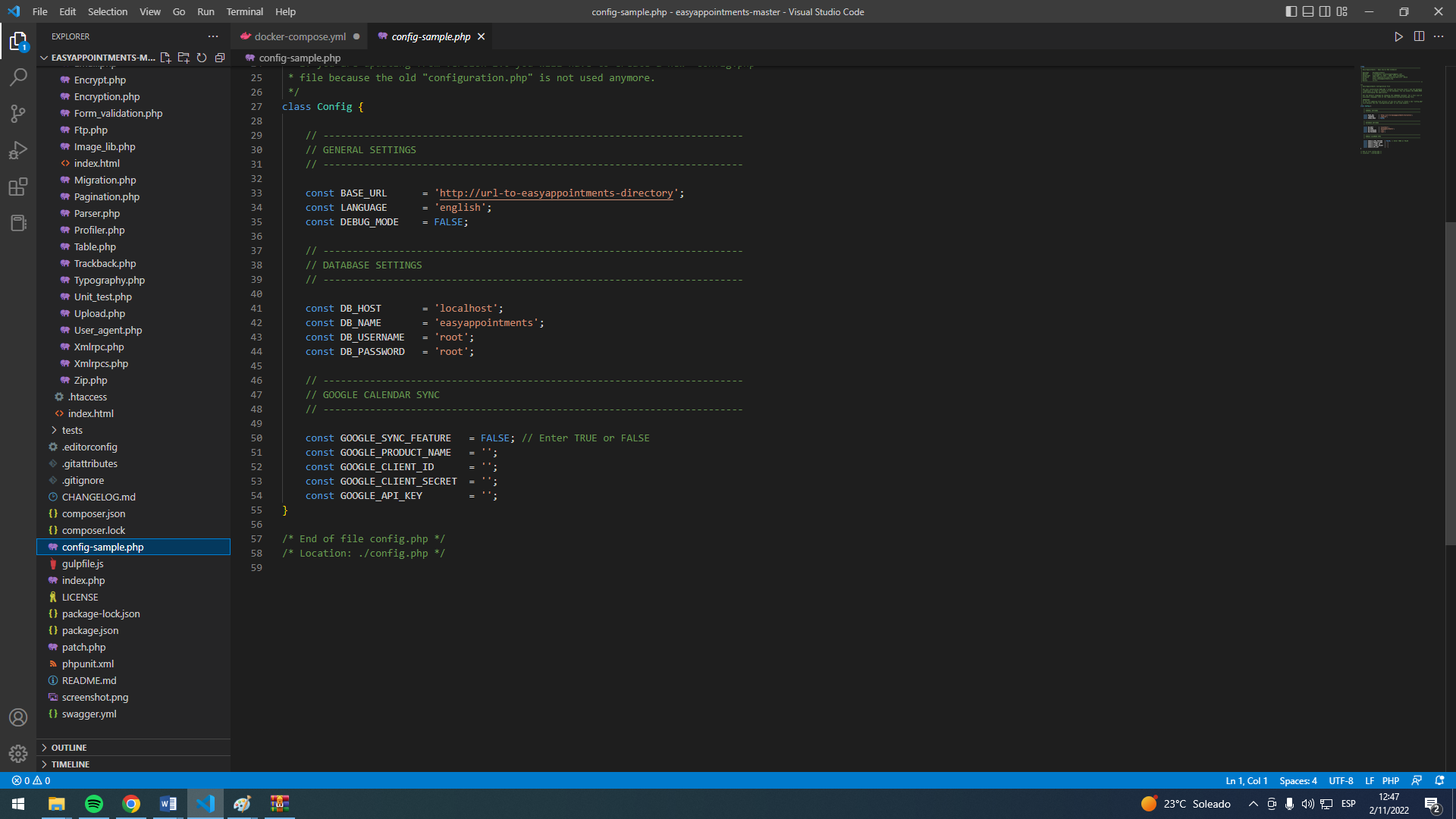


Para el logout:



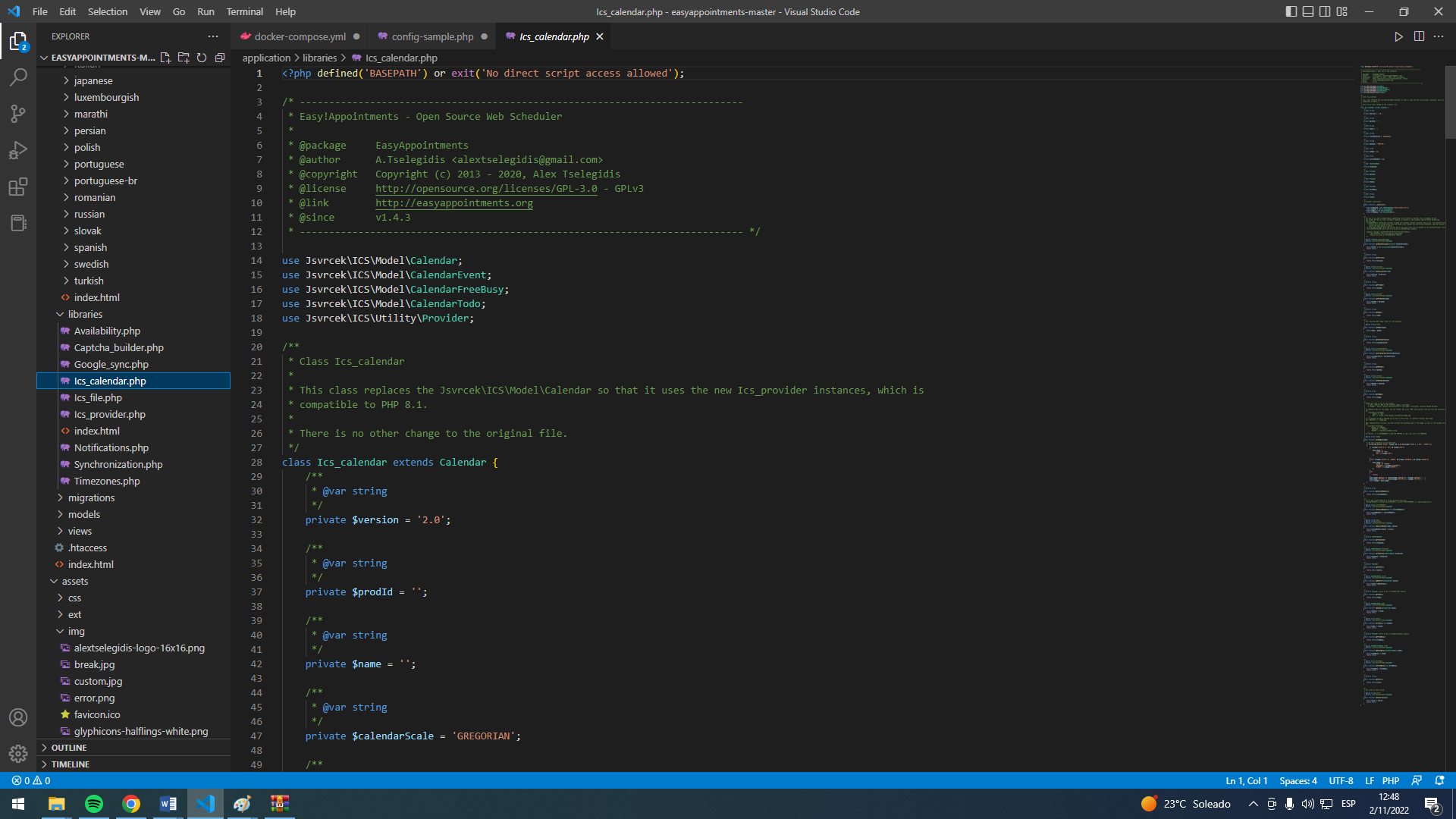
Clase Config

Definimos la clase Config:



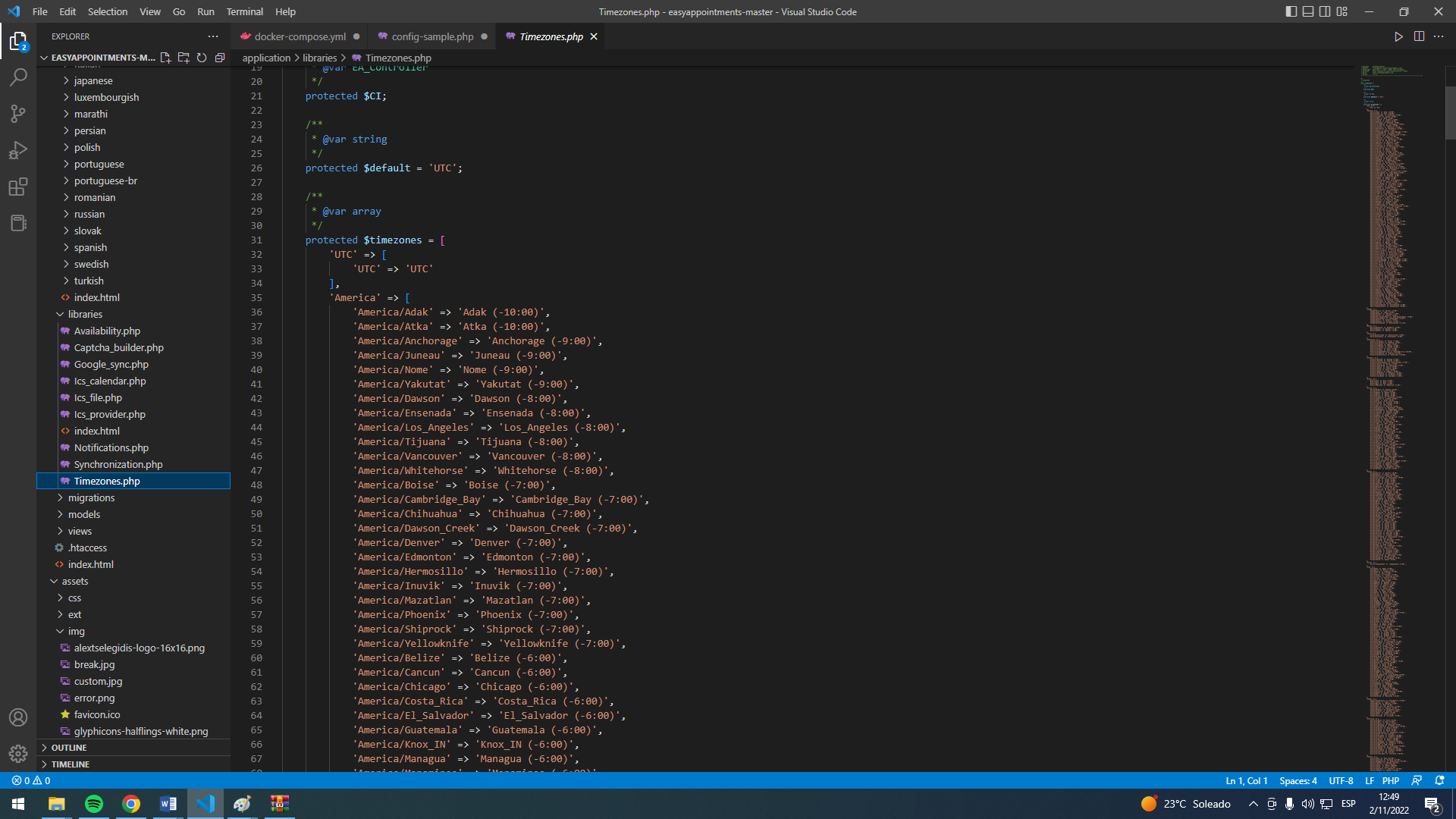
## Importación de Calendario:

Definimos el calendario a utilizar:



Zona horaria

También la zona horaria:

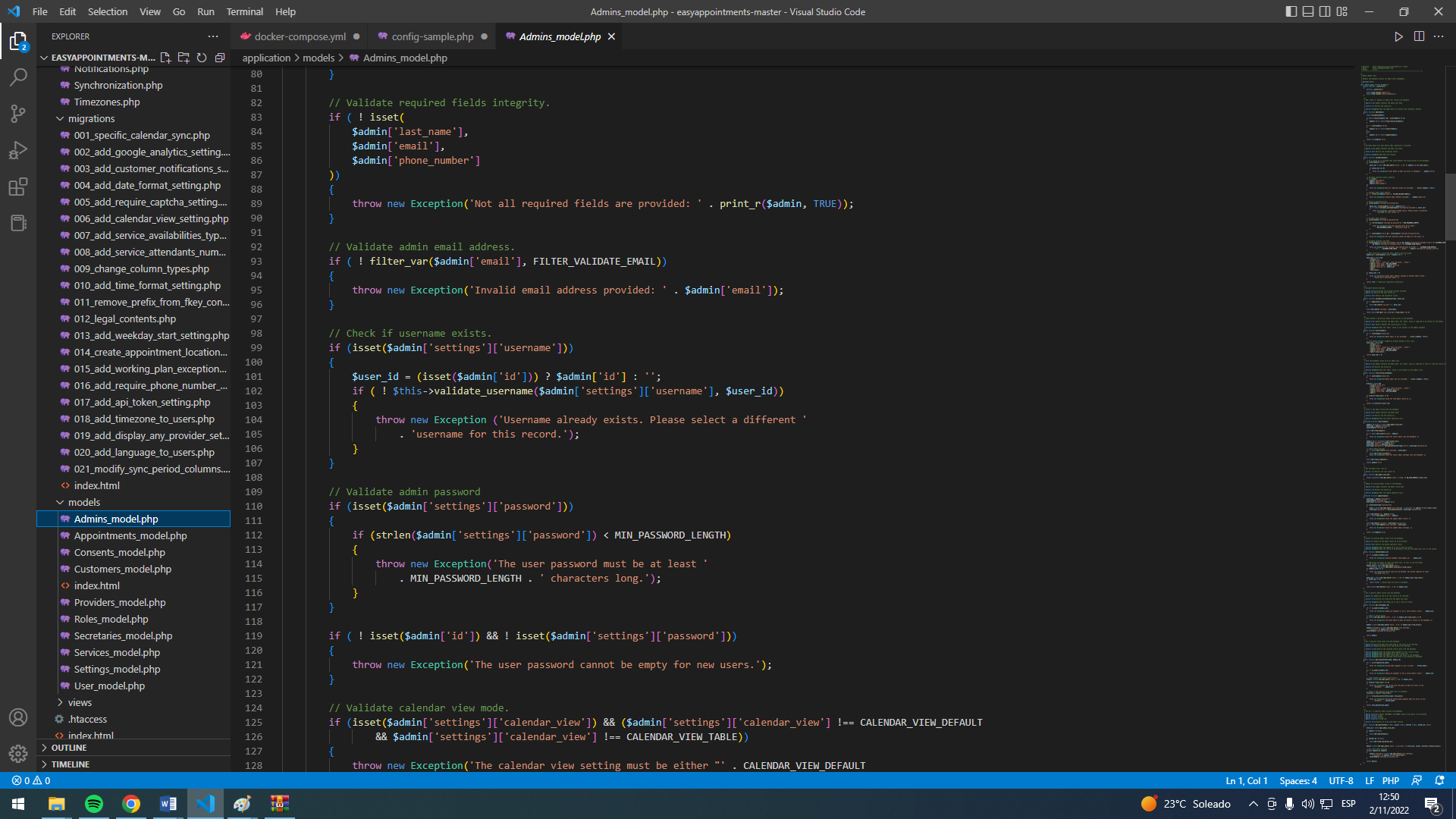


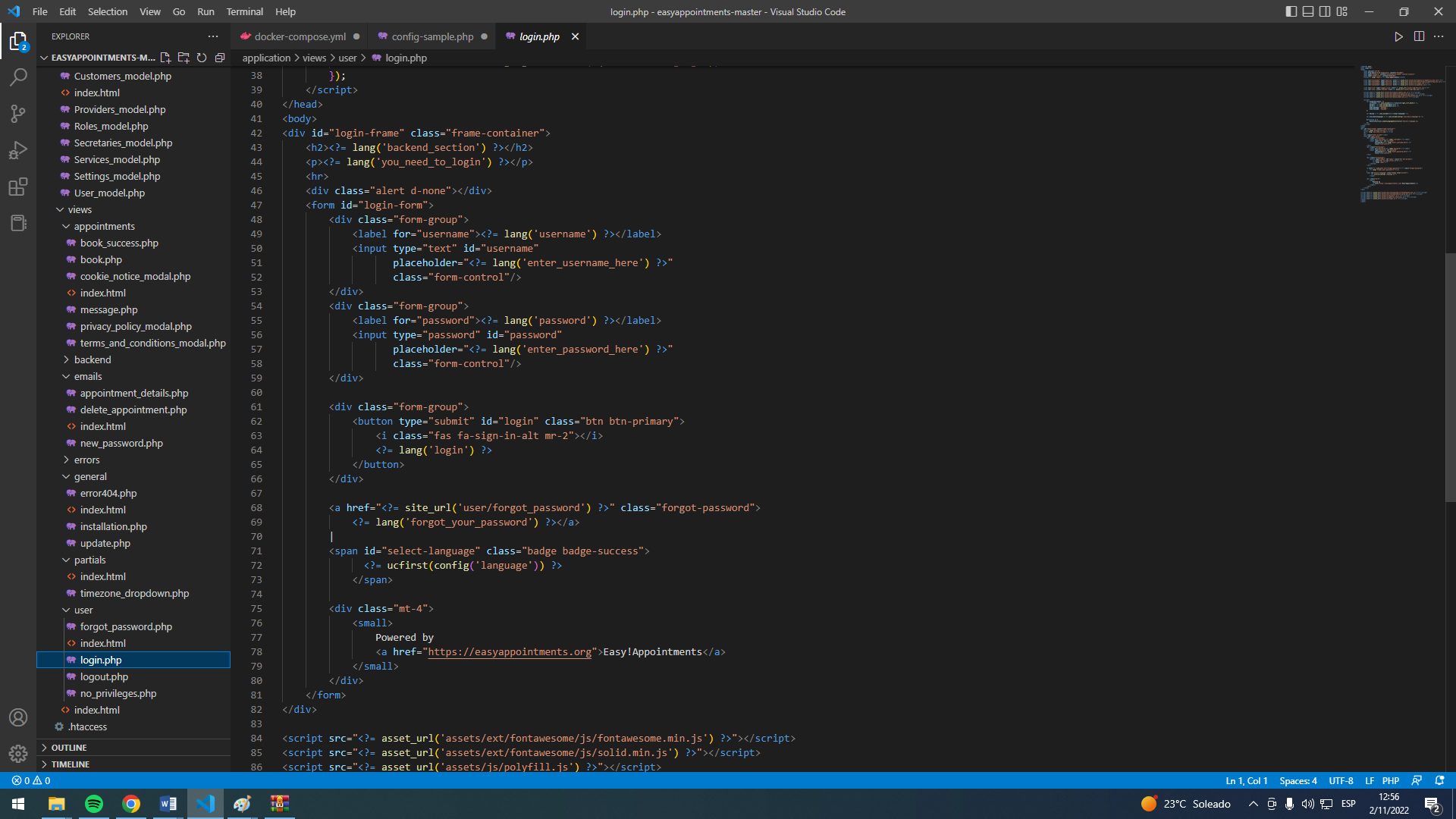
Bajo esta configuración al momento de registrar una reserva podremos observar un listado de horarios y días disponibles como ayuda rápida:



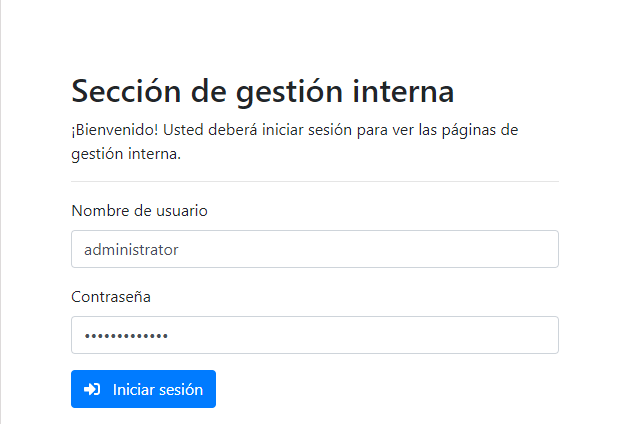
Clase Admin

Ahora configuramos la clase admin que tendrá el control de los registros que se realizaran en la aplicación:



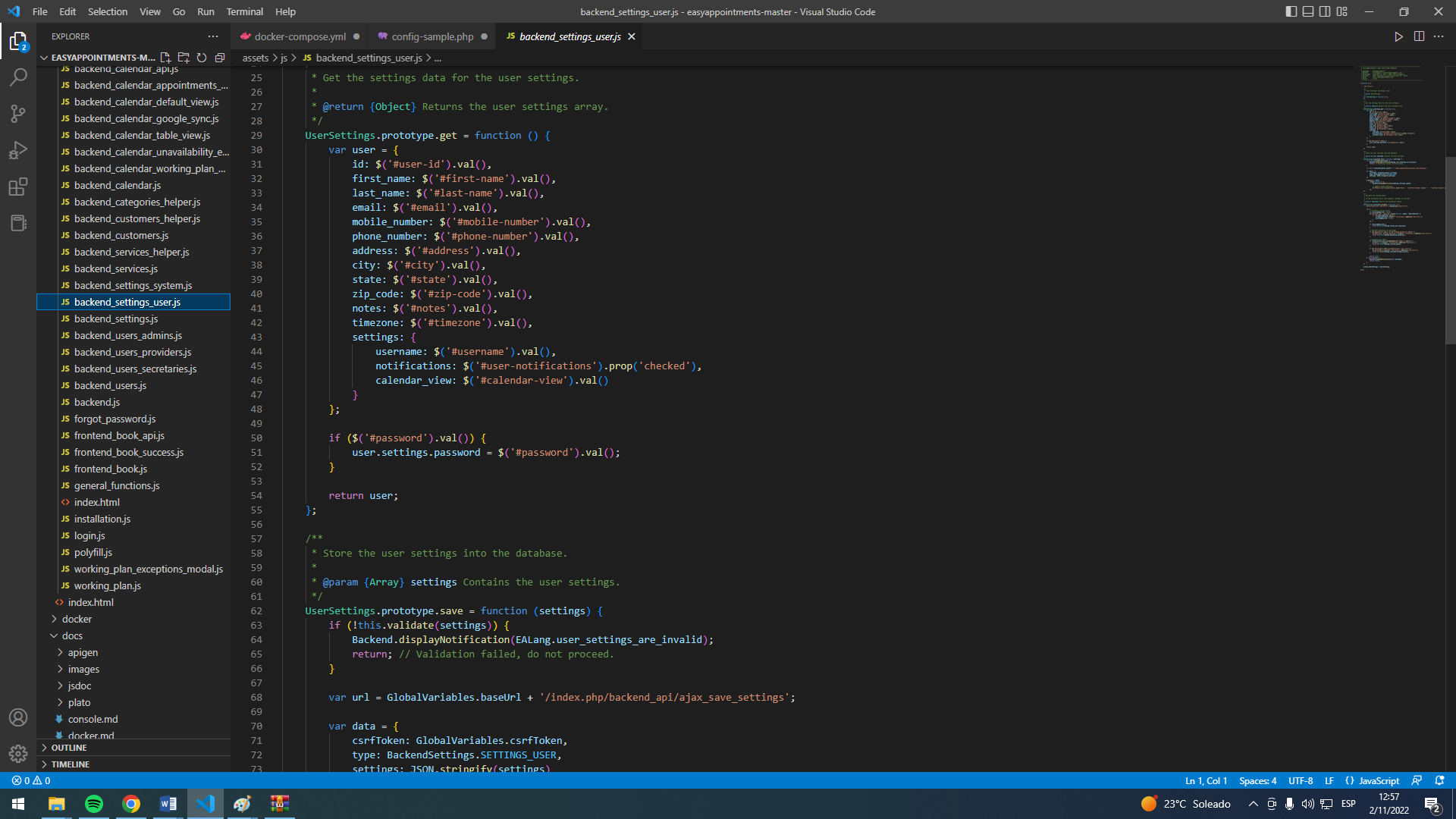


Una vez terminado la programación de las líneas de código tendremos este primer resultado:

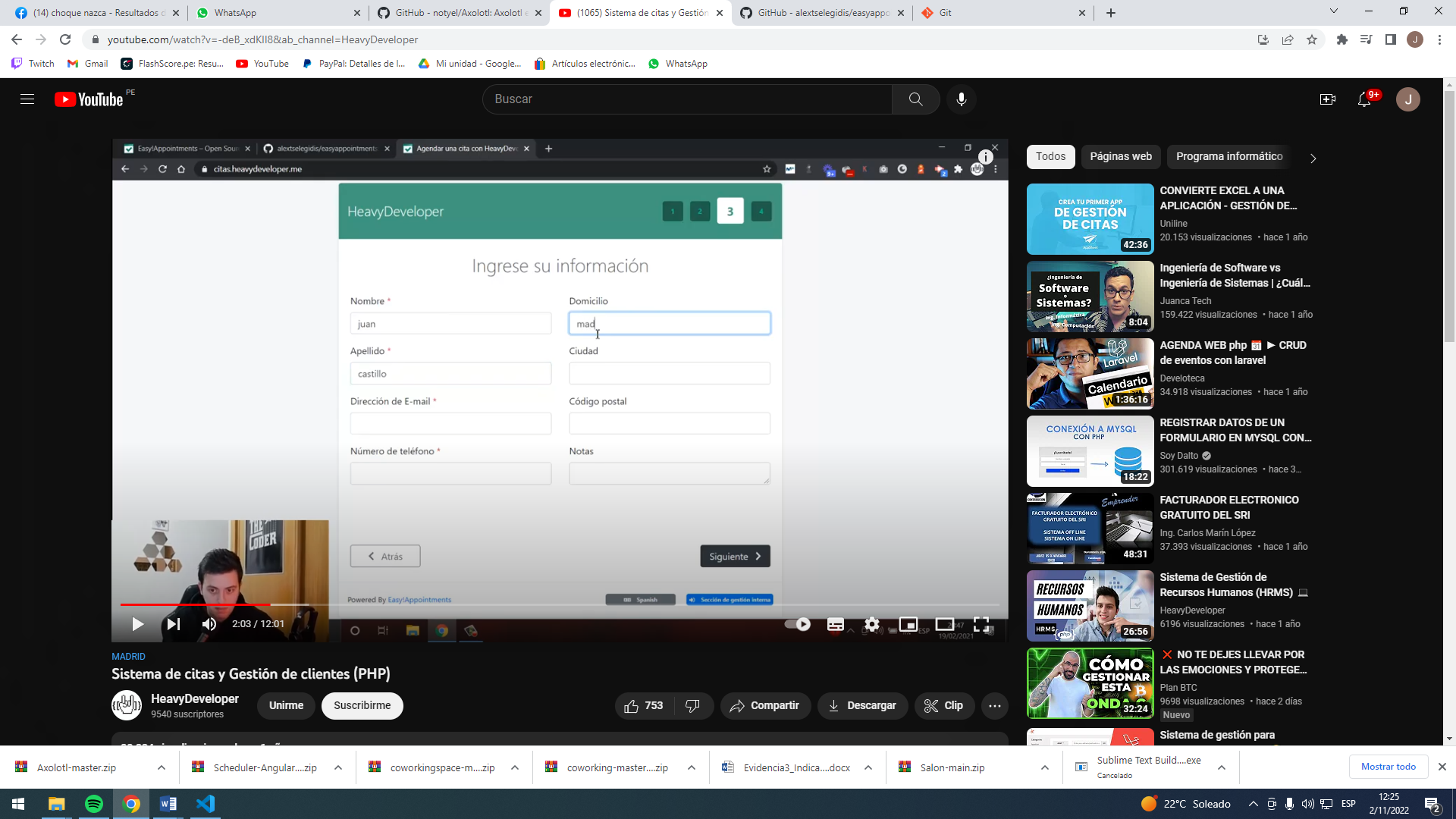


Clase Usuario

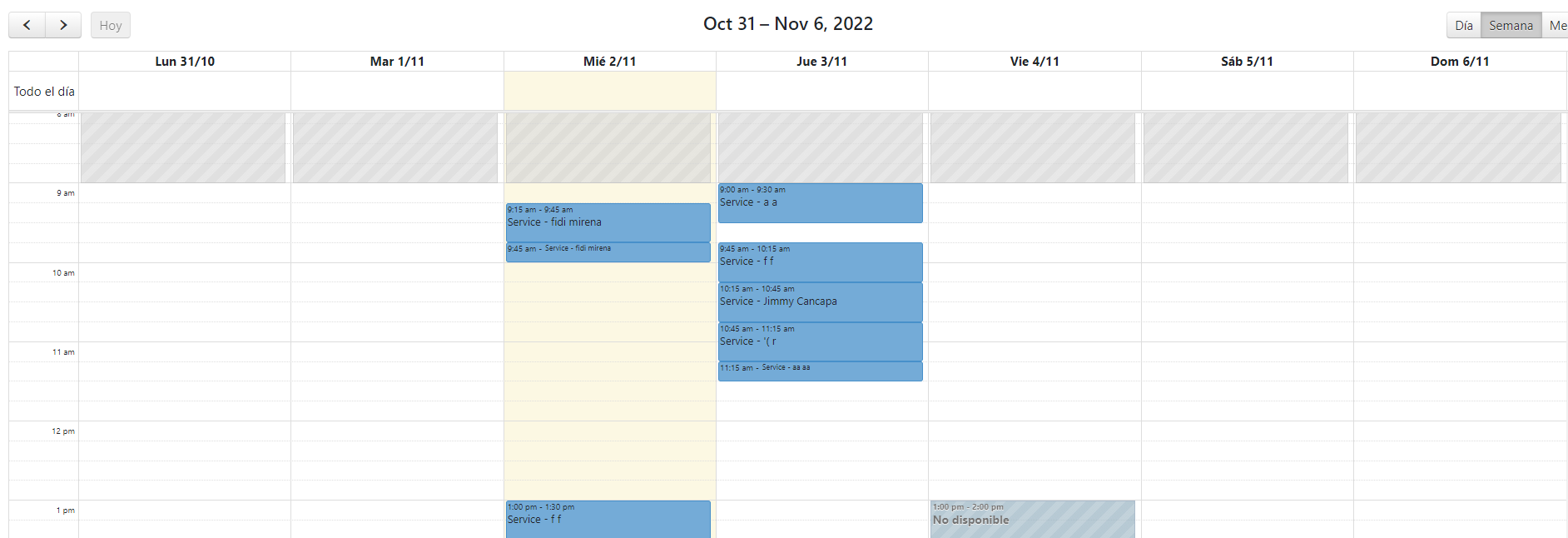
Ahora pasamos a configurar la sección de usuarios donde podrán registrar los datos respectivos para la reservación que soliciten



Este es el resultado luego de la línea de comandos establecida en Visual Studio Code:



Una vez importado el calendario y programado correctamente la clase usuario podremos visualizar en el proyecto todas las citas que se realizaron en el sistema:



Finalmente, si hacemos un clic en alguna hora libre, se tendrá la posibilidad de registrar una nueva reserva abriendo una hoja de llenado de datos de usuario, cuya programación se explicó en la clase Usuario:

