

PROYECTO FINAL DE CICLO

MEMORIA EXPLICATIVA



Damián Millán Rivero

2ºDAM



ÍNDICE

1. INTRODUCCIÓN Y MOTIVACIÓN
2. DESCRIPCIÓN O RESUMEN GENERAL DEL PROYECTO
3. DESCRIPCIÓN DE LOS DIFERENTES ESTADOS DE DESARROLLO DEL PROYECTO CON EL DETALLE TÉCNICO.
 - 3.1 PLANIFICACIÓN
 - 3.2 ANÁLISIS
 - 3.3 DISEÑO DE LAS APLICACIONES Y LA BASE DE DATOS
 - 3.4 IMPLEMENTACIÓN
 - 3.5 FUTURAS FUNCIONALIDADES
4. PRUEBAS DE FUNCIONAMIENTO EFECTUADAS
5. CONCLUSIÓN Y VALORACIÓN PERSONAL
6. BIBLIOGRAFÍA Y RECURSOS



1. INTRODUCCIÓN Y MOTIVACIÓN

Basándome en las necesidades del centro educativo en el que estudio, he creado una aplicación que gestiona los registros de la jornada laboral y el horario de los empleados de un centro educativo o empresa mediante aplicación de escritorio y móvil.

Las aplicaciones se conectarán con un servidor local del centro, y una vez conectadas, se podrán realizar los fichajes y consultar el horario semanal.

He creado estas aplicaciones con el fin de agilizar el proceso de entrada y salida que actualmente se sigue realizando mediante firma manuscrita. Esto ocasiona aglomeraciones en la entrada del centro y supone una pérdida de tiempo que se podría evitar con estas aplicaciones.

2. DESCRIPCIÓN O RESUMEN GENERAL DEL PROYECTO

He desarrollado una aplicación móvil (Android e IOS) con FLUTTER (framework SDK multiplataforma de interfaz de usuario móvil gratuito y de código abierto. Esto permite a los desarrolladores utilizar una única base de código para las aplicaciones iOS y Android.) que solicite usuario y contraseña del empleado y que una vez entre, permita fichar las horas de trabajo del día, también permite visualizar el horario semanal, nos informará de cuando hicimos el último fichaje y recibir notificaciones, por ejemplo, de aviso en caso de tener fichajes olvidados, etc.



Posteriormente, he diseñado una aplicación GUI (Graphical User Interface) que también requiere de usuario y contraseña para acceder a ella. Basándose en el control de entrada diario de la base de datos permite tener un listado de los empleados que han fichado o no en un día determinado, así podremos saber que compañeros han faltado al trabajo de una forma rápida y eficaz. Además permite al usuario tener un listado de las aulas disponibles y puede visualizar las que están ocupadas y por supuesto te permite fichar al igual que en la aplicación móvil, así el usuario tiene más de una opción de completar sus fichajes. También se mostrarán notificaciones de última conexión y de fichajes olvidados.

Para el correcto funcionamiento de estas aplicaciones es necesario un servidor de aplicaciones en la intranet para poder hacer peticiones a la API, la cual ha sido creada e implementada con Spring Boot en JAVA.

Por último he creado un programa JAVA que, basándose en un fichero XML de usuarios, los da de alta en la base de datos de gestión del sistema. Esto servirá para poder dar de alta al profesorado del centro leyendo un fichero XML proporcionado

3. DESCRIPCIÓN DE LOS DIFERENTES ESTADOS DE DESARROLLO DEL PROYECTO CON EL DETALLE TÉCNICO.

3.1 PLANIFICACIÓN

Antes de realizar este proyecto tuve que realizar un estudio de viabilidad el cual resultó positivo ya que es una necesidad en todas las empresas. El número total de

empresas registradas en España según la actualización del Directorio Central de Empresas (DIRCE) a fecha 13/12/2021 es de 3 366 570.

Tras analizar los costes planifiqué el tiempo estimado poniendo como fecha límite unas semanas antes de la entrega del proyecto para tener un margen en caso de fallos o errores que no hubiese previsto y que no permitieran funcionar correctamente las aplicaciones.

3.2 ANÁLISIS

API REST con Spring Boot:

- Es la encargada de hacer las peticiones a la base de datos mediante los métodos CRUD (Create, Read, Update and Delete).

```

41 }
42
43
44 @Operation(summary="Obtiene una lista de horarios por el día de la semana")
45 @ApiResponse(value = {
46     @ApiResponse(responseCode = "600", description = "El horario existe",
47         content = @Content(schema = @Schema(implementation =
48             Horario.class))),
49     @ApiResponse(responseCode = "404", description = "El horario no existe",
50         content = @Content(schema = @Schema(implementation =
51             Horario.class)))
52 })
53 @GetMapping(value = "/horario/diaSemana/{diaSemana}", produces = "application/json")
54 public ResponseEntity<List<Horario>> getHorarioDiaSemana(@RequestParam(value =
55     "diaSemana", defaultValue = "") int diaSemana) {
56     List<Horario> horarios = null;
57     horarios = horarioService.findByDiaSemana(diaSemana);
58     return new ResponseEntity<>(horarios, HttpStatus.OK);
59 }
60

```

```

65 Dim headers = New List(Of Parametro)
66 Dim parametros = New List(Of Parametro)
67
68 Dim response = api.MGet(url, headers, parametros)
69
70 Dim fichas As List(Of Ficha) = JsonConvert.DeserializeObject(Of List(Of Ficha))(response)
71
72 Return fichas
73 End Function
74
75 0 referencias
76 Function horaHorario(diaHoy As Integer) As List(Of Horario)
77     Dim api = New DBApi
78     Dim url As String = "http://localhost:8080/horario/diaSemana/{diaSemana}?diaSemana=" & diaHoy
79     Dim headers = New List(Of Parametro)
80     Dim parametros = New List(Of Parametro)
81
82     Dim response = api.MGet(url, headers, parametros)
83
84     Dim horarios As List(Of Horario) = JsonConvert.DeserializeObject(Of List(Of Horario))(response)
85
86 Return horarios
87 End Function

```

Implementación de un método que busca una ficha por el id del horario y si no existe la crea.

```
44 public Ficha findByIdHorario(long id) {  
45     Set<Ficha> fichas = fichaRepository.findAll();  
46     Ficha fichaIdHorario = new Ficha();  
47     boolean noExiste = false;  
48     for (Ficha f:fichas) {  
49         if (f.getHorario().getIdHorario() == id && f.getFecha().equals(LocalDate.now())) {  
50             fichaIdHorario = f;  
51             noExiste = true;  
52         }  
53     }  
54     if (!noExiste) {  
55         Optional<Horario> horarioOpt = horarioRepository.findById(id);  
56         if (horarioOpt.isPresent()) {  
57             fichaIdHorario.setHorario(horarioOpt.get());  
58             fichaIdHorario.setFichado(false);  
59             fichaIdHorario.setFecha(LocalDate.now());  
60             fichaRepository.save(fichaIdHorario);  
61         }  
62     }  
63     return fichaIdHorario;  
}
```

La aplicación GUI tiene que:

- Almacenar las entradas de los distintos empleados en una base de datos haciendo un login del usuario en el sistema y seleccionando los fichajes diarios.

**IES MARCOS ZARAGOZA** Centre digital
col·laboratiu

Usuario (DNI)

Contraseña



- Notificar al usuario la última conexión haciendo una petición a la base de datos de los fichajes de la última semana.



- Mostrar el horario recibiendo desde la base de datos los fichajes semanales, de un trabajador.

frmHorario

 **IES MARCOS ZARAGOZA**
Usuario: damian millan


<---


	Lunes	Martes	Miercoles	Jueves	Viernes
08:00 - 08:55	----	----	----	----	inf
08:55 - 09:50	----	----	----	----	----
09:50 - 10:45	----	----	----	MECAINFO	----
10:45 - 11:10	----	----	----	----	----
11:10 - 12:05	----	A07	----	----	----
12:05 - 13:00	A11	----	----	----	A20B
13:00 - 13:55	----	----	----	----	----
13:55 - 14:15	----	----	----	----	----
14:15 - 15:30	----	----	----	----	----
15:30 - 16:25	----	----	----	----	----
16:25 - 17:20	----	APROY	----	----	----
17:20 - 18:15	----	----	----	----	----
18:15 - 18:35	----	----	----	----	----
18:35 - 19:30	----	----	----	----	----
19:30 - 20:25	----	----	----	----	A29B
20:25 - 21:20	----	----	----	----	----

*

- Mostrar un listado de los trabajadores que han faltado al trabajo(o que se han olvidado de fichar) mediante una petición a la base de datos de los fichajes diarios de los compañeros.

frmBuscar

 **IES MARCOS ZARAGOZA**
Usuario: damian

 Centre digital col·laboratiu

<---

☐ Profesor
 ☐ Aula
 ☒ Profesores no fichados

Nombre

Fecha

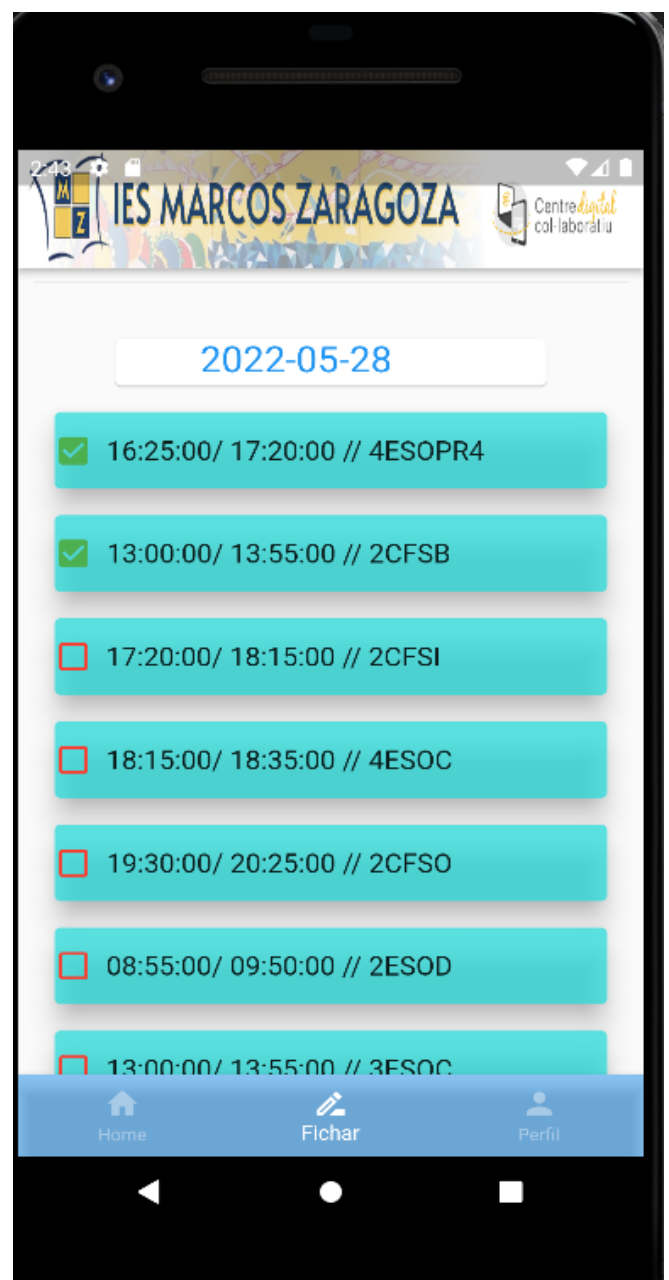
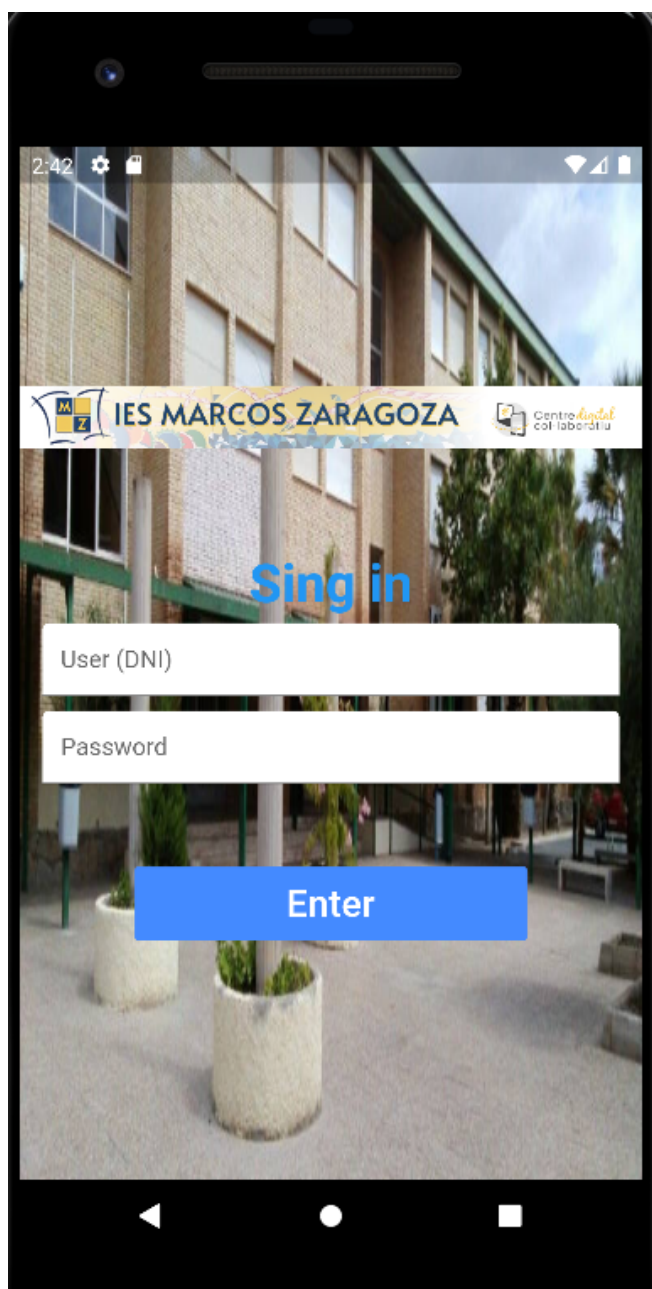
Hora Inicio (hh:mm)

	Nombre	Apellidos	Estado
▶	damian	millan	Presentado
	Lucena	Tamara del	No presentado
	Riquelme	Reynaldo Girona	No presentado
	Cuevas	Adelia Naranjo	No presentado
	Llorens	Tadeo Quirino	No presentado
	Tirado	Xiomara Esteve	No presentado
	Madrigal	Isaac Ribas	No presentado
	Pombo	Emperatriz Marti	No presentado

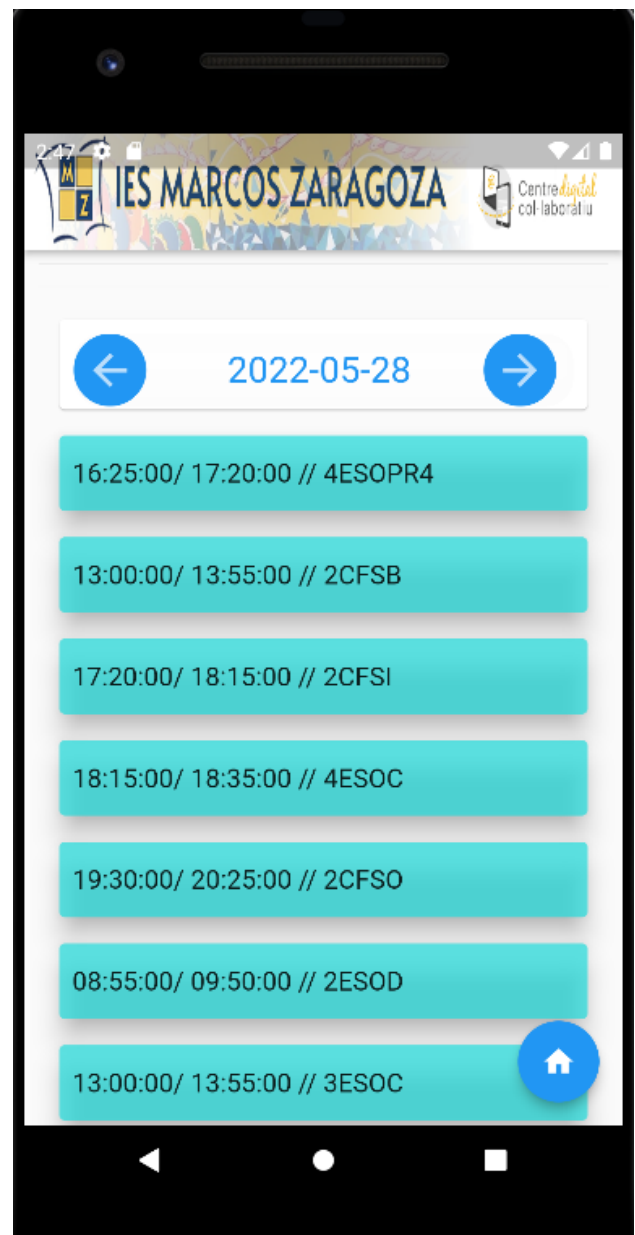
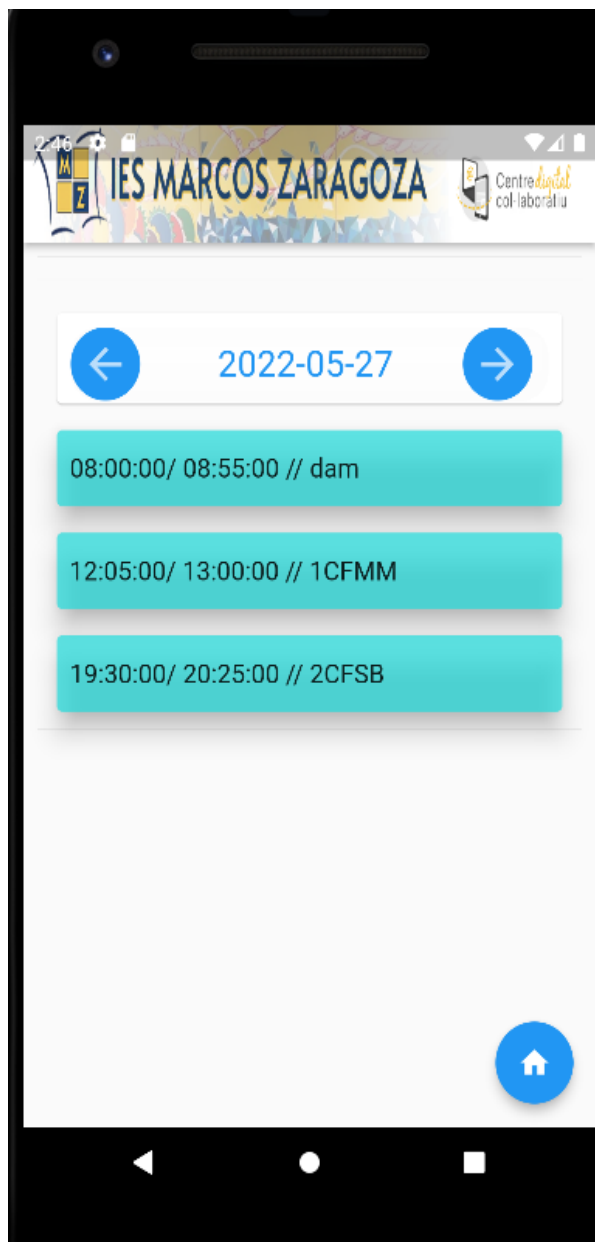
- Mostrar un listado de las aulas libres con una petición a los horarios diarios y descartando todas las aulas ocupadas en la hora seleccionada.

La aplicación móvil tiene que:

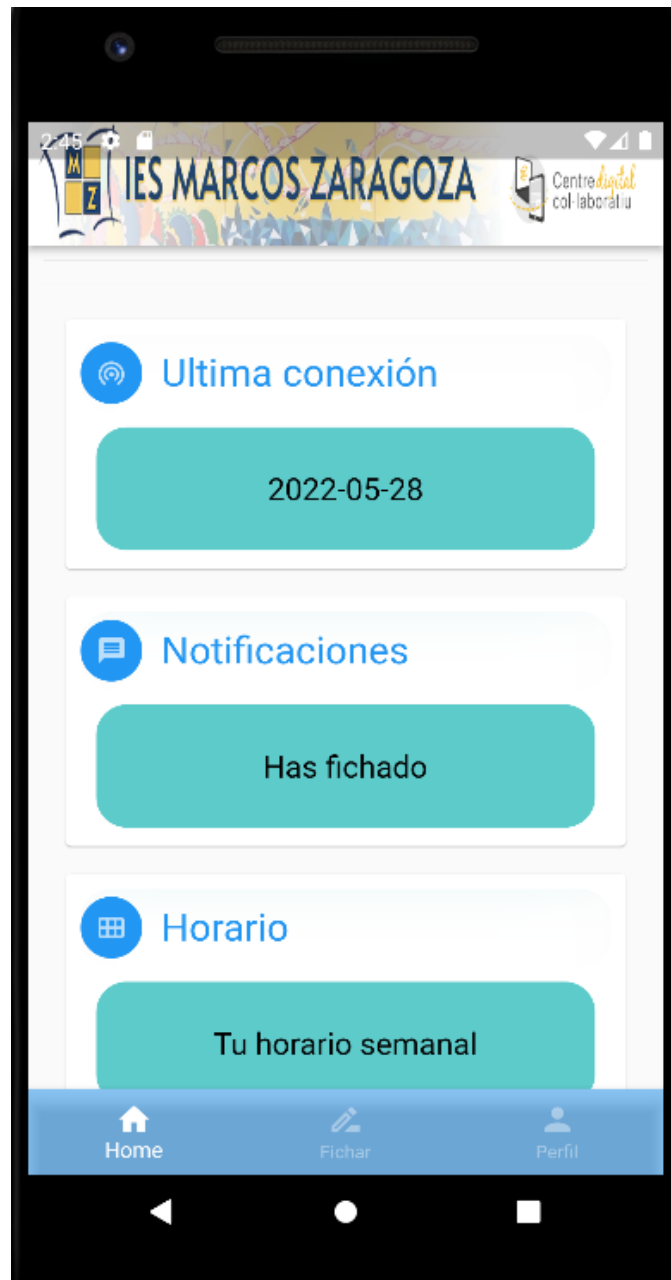
- Almacenar las entradas de los distintos empleados en una base de datos haciendo un login del usuario en el sistema y seleccionando los fichajes diarios.



- Mostrar el horario recibiendo desde la base de datos los fichajes semanales de un trabajador.



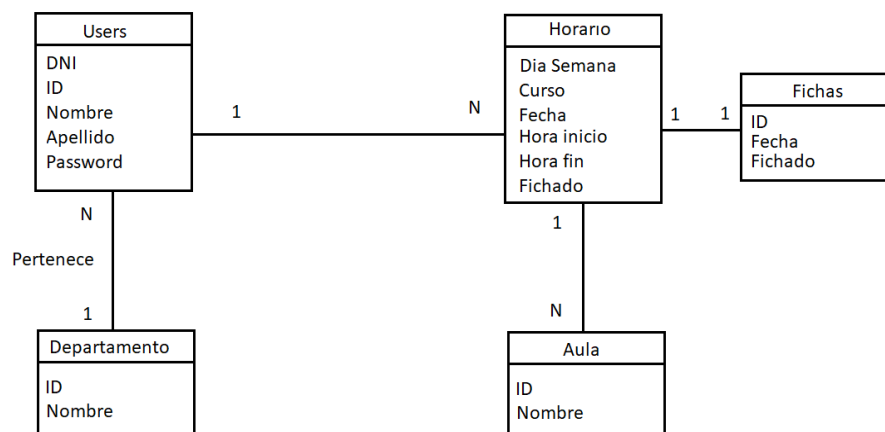
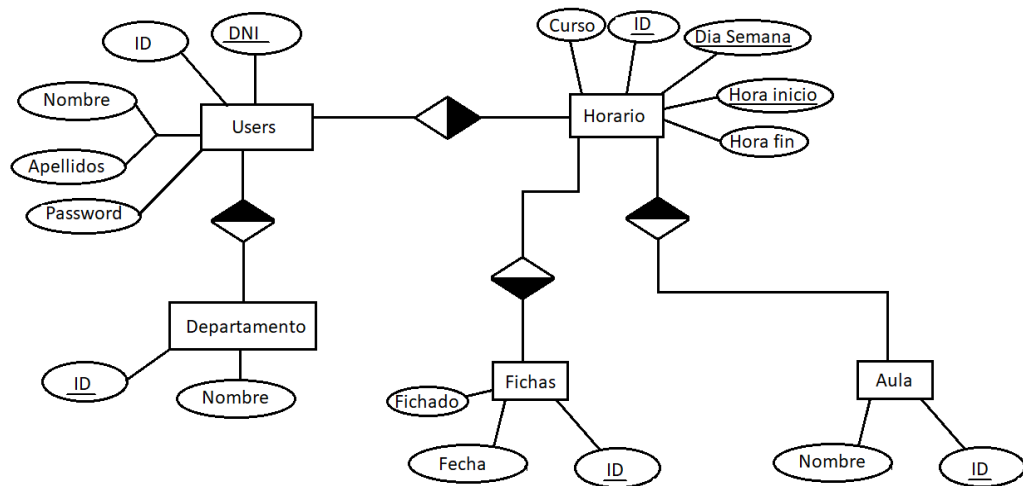
- Notificar al usuario la última conexión haciendo una petición a la base de datos de los fichajes de la última semana.
- Notificar al usuario si ha fichado o no hoy mediante una petición a la base de datos del horario diario.



3.3 DISEÑO DE LAS APLICACIONES Y LA BASE DE DATOS

Después de rediseñar las aplicaciones varias veces a mano, finalmente el diseño escogido es el siguiente:

Base de datos



Aplicación GUI

Ventana Login

IES Marcos Zaragoza

Icono empresa

nombreUsuario

Contraseña

Entrar

Ventana Principal

IES Marcos Zaragoza

Usuario: NombreUsuario

Fichar Solicitar

Horario Cerrar sesión

Buscar

Ultima conexión
*HH D/M/A

Ventana Fichar

IES Marcos Zaragoza

Usuario: NombreUsuario

dd/mm/aaaa

Hora ini / Hora fin/ Aula Fichar

Hora ini / Hora fin/ Aula Fichar

Hora ini / Hora fin/ Aula Fichar

Hora ini / Hora fin/ Aula Fichar

Ventana Horario

IES Marcos Zaragoza

Usuario: NombreUsuario

hh -- -- Aula Aula Aula

hh Aula Aula Aula Aula Aula

hh Aula Aula Aula Aula Aula

hh -- -- Aula Aula Aula

hh Aula Aula -- -- Aula

hh Aula -- -- -- --

Ventana Buscar

IES Marcos Zaragoza

Usuario: NombreUsuario

Profesor ☒ Aula ☐

Nombre

Hora

Estado

Buscar

Aplicación móvil

IES Marcos Zaragoza

Login

Conexión electrónica

Contraseña

Entrar

IES Marcos Zaragoza

Estado

Última conexión (último fichaje)

Notificaciones

Fichaje olvidado

Horario

HORARIO

Historial de fichajes

HORARIO

Buscar Fichar

IES Marcos Zaragoza

Fichar

Hora / Aula / Curso Fichado

Hora / Aula / Curso Fichado

Hora / Aula / Curso Fichado

Hora / Aula / Curso Fichado

Hora / Aula / Curso Fichado

Hora / Aula / Curso Fichado

Estado

Última conexión (último fichaje)

IES Marcos Zaragoza

Historial

Día / Mes / Año

Hora inicio / Hora fin Fichado

Hora inicio / Hora fin Fichado

Hora inicio / Hora fin Fichado

Hora inicio / Hora fin Fichado

Hora inicio / Hora fin No Fichado

Día / Mes / Año

Hora inicio / Hora fin No Fichado

Hora inicio / Hora fin Fichado

Hora inicio / Hora fin Fichado

Hora inicio / Hora fin Fichado

Hora inicio / Hora fin Fichado

3.4 IMPLEMENTACIÓN

Los IDE seleccionados para el desarrollo de las aplicaciones han sido Visual Studio e IntelliJ. Otra de las herramientas utilizadas para el desarrollo de esta aplicación es la Base de Datos MySQL.

Los lenguajes utilizados han sido:

- DART con el framework Flutter para la aplicación móvil
- Visual Basic para la aplicación GUI
- JAVA para Spring Boot y lector de ficheros

3.5 FUTURAS FUNCIONALIDADES

Una de las implementaciones a llevar a cabo en un futuro sería la solicitud de fichaje olvidado. Esta funcionalidad consiste en que el programa detecta que has olvidado un fichaje y envía una notificación al usuario, este debe solicitar mediante una petición por la aplicación al departamento de RRHH o al responsable de su departamento la modificación de este fichaje olvidado.

La primera vez en hacer el login tendrá que mostrar una ventana nueva donde se hará el cambio de contraseña por seguridad ya que todas las contraseñas iniciales son una contraseña estándar.

4. PRUEBAS DE FUNCIONAMIENTO EFECTUADAS

Aplicación móvil:

❖ Login page:

- usuario y contraseña vacías.
- usuario correcto contraseña vacía.
- usuario incorrecto contraseña vacía.
- usuario vacío contraseña correcta.
- usuario vacío contraseña incorrecta.
- usuario correcto contraseña incorrecta.
- usuario incorrecto contraseña correcta.
- usuario correcto contraseña correcta.

❖ Home:

- Fichajes vacíos.
- Fichajes todos marcados.
- Fichajes sin marcar.

❖ Fichajes:

- Sin fichajes.
- Lista de 10 fichajes.
- Marcar todos los fichajes.
- No marcar ningún fichaje.

❖ Perfil:

- Botón cerrar sesión y comprobar que no se guardan los datos del usuario (dni y contraseña).

❖ Horario:

- Sin fichajes.

- Fichajes solo el día a probar.
- Fichajes todos los días menos el día a probar.

Aplicación GUI:

❖ Form login:

- usuario y contraseña vacías.
- usuario correcto contraseña vacía.
- usuario incorrecto contraseña vacía.
- usuario vacío contraseña correcta.
- usuario vacío contraseña incorrecta.
- usuario correcto contraseña incorrecta.
- usuario incorrecto contraseña correcta.
- usuario correcto contraseña correcta.

❖ Form principal:

- Botón cerrar sesión y comprobar que no se guardan los datos del usuario (dni y contraseña).
- Notificaciones fichajes vacíos.
- Notificaciones fichajes del día anterior efectuados.
- Notificaciones fichaje del día de prueba efectuados.
- Comprobación de las rutas de los botones al hacer clic en ellos.

❖ Form fichar:

- Sin fichajes.
- Lista de 10 fichajes.
- Marcar todos los fichajes.
- No marcar ningún fichaje.



❖ Form horario:

- Sin fichajes.
- Fichajes solo el día a probar.
- Fichajes todos los días menos el día a probar.

❖ Form buscar:

- Intento de marcar dos opciones.
- Profesor marcado, nombre vacío, fecha actual.
- Profesor marcado, nombre vacío, fecha anterior.
- Profesor marcado, nombre incorrecto, fecha actual.
- Profesor marcado, nombre correcto, fecha actual.
- Profesor marcado, nombre incorrecto, fecha anterior.
- Profesor marcado, nombre correcto, fecha anterior.
- Aula marcada, nombre vacío, todas las horas.
- Aula marcada, nombre incorrecto, todas las horas.
- Aula marcada, nombre correcto, todas las horas.
- Profesores no fichados marcado, varios días de la semana.

5. CONCLUSIÓN Y VALORACIÓN PERSONAL

Con este proyecto he aprendido a programar en Visual Basic y he mejorado mucho mis conocimientos en Flutter. Después de diversas consultas, he aprendido a buscar información y aplicarla según las necesidades del proyecto.

Creo que es un proyecto muy útil tanto para empresas como para trabajadores ya que ayuda a tener un orden y control en tu horario. En mi opinión sería un programa completamente viable para este centro teniendo en cuenta la situación actual.



6. BIBLIOGRAFÍA ESTILO VANCOUVER Y RECURSOS

- Stack Overflow en español [Internet]. Stack Overflow en español. Disponible en: <https://es.stackoverflow.com/>
- Before you continue to YouTube [Internet]. Youtube; Disponible en: <https://www.youtube.com/c/FernandoHerreraCr>
- Dart packages [Internet]. Dart packages. Disponible en: <https://pub.dev/>
- AULES: Inicia sessió en este lloc [Internet]. Gva.es. Disponible en: <https://aules.edu.gva.es/fp/my/>
- Dart [Internet]. Stack Secrets. 2020. Disponible en: <https://stacksecrets.com/dart/>
- Code utility [Internet]. Code Utility. Disponible en: <https://codeutility.org/>
- dotnet-bot. Documentos de Visual Basic: inicio, tutoriales y referencias [Internet]. Microsoft.com. Disponible en: <https://docs.microsoft.com/es-es/dotnet/visual-basic/>
- Howtos de Java [Internet]. Delft Stack. Disponible en: <https://www.delftstack.com/es/howto/java/>