

María Monteagudo Robledillo
2º DAW

índice

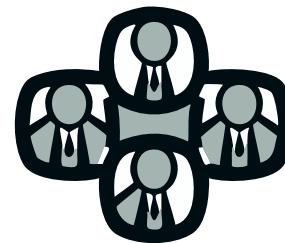
1. Contextualización	2
2. Objetivo	3
3. Análisis de mercado	4
4. Metodologías usadas	5
5. Tecnologías usadas	7
6. Base de datos	10
7. Back end	12
8. Problemas y dificultades	19
9. Resultados obtenidos	21
10. Líneas futuras	24
11. Conclusión	9
12. Bibliografía	10

1. Contextualización

El proyecto Vehoclinic es una innovadora solución desarrollada por más de seis años por la empresa Clase 10, en colaboración con un grupo de clínicas veterinarias repartidas por toda España



Gestión logística y financiera.



Gestión empresarial.

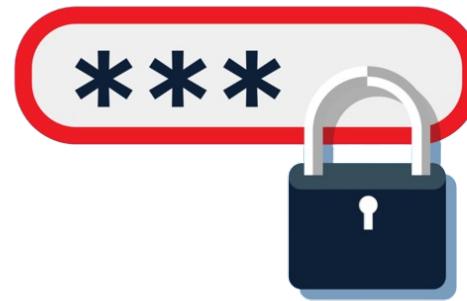
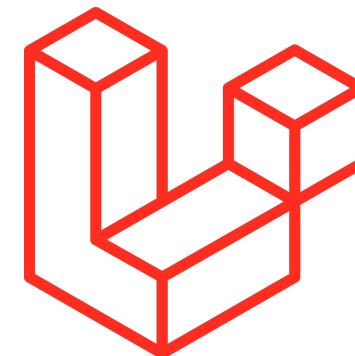
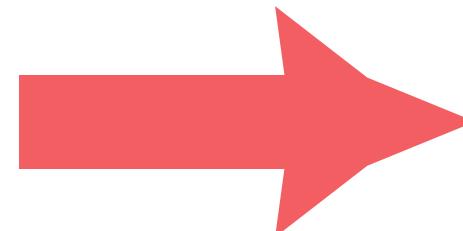
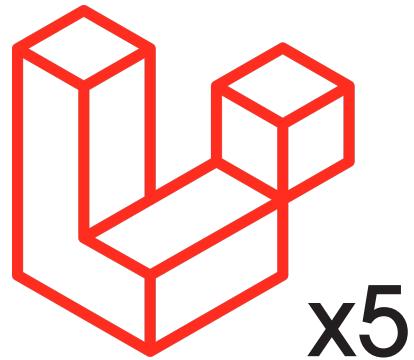


Gestión clínica y peluquería.



Movilidad y conexión entre las apps

2. Objetivos



3. Análisis de mercado

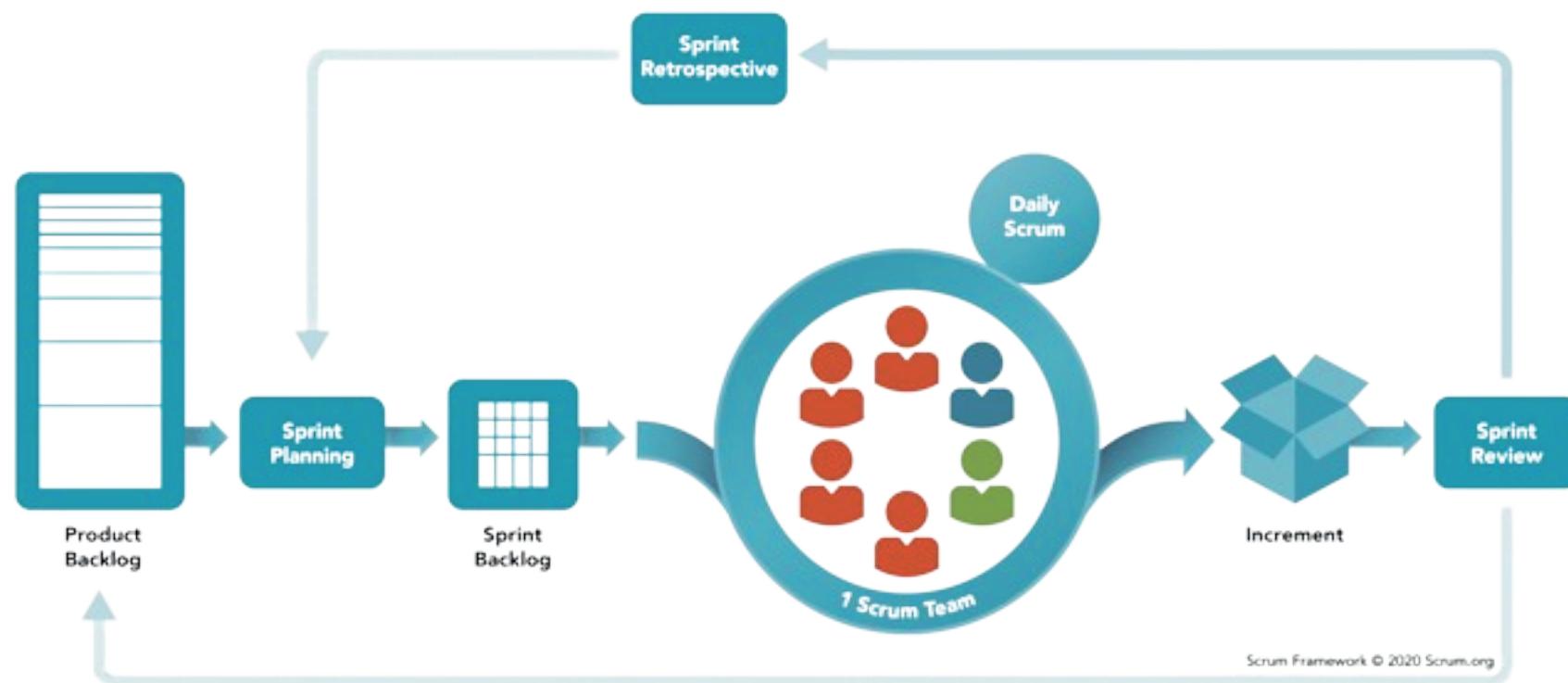
Gestión clínica y hospitalaria veterinaria:

- Registro y seguimiento.
- Administración de agendas y citas.
- Gestión de clientes.
- Revisiones.
- Especialidad para gestionar casos.
- Medicina preventiva
- Gestión de solicitudes.
- Control y seguimiento de servicios
- Funcionalidades para gestionar la hospitalización.

Gestión empresarial:

- Configuración personalizada
- Busqueda avanzada y básica
- Generación de informes
- Informes predefinidos
- Firma digital
- Herramientas de marketing y comunicación

4. Metodologías usadas



Metodología SCRUM

4. Metodologías utilizadas

The screenshot shows the PARTES10 software interface. At the top, there are three buttons: 'Partes' (highlighted in pink), 'Procesos', and 'Resumen'. The title bar says 'PARTES10' and 'COLABORADOR: 217 - MARIA MONTEAGUDO'. On the right, it shows 'PARTES 10 - Versión 20230223', 'FECHA: 03/04/2024 HORA: 17:14:50', and three icons: 'Control Horario', 'Historial y Sel.', and 'Ficha de trabajo'.

Below the title bar, there are two dropdown menus: 'Tipo de proceso:' (with options 'Todos', 'Deseleccionar', and 'General - Implantación') and 'Estado:' (with options 'Todos', 'Deseleccionar', and a long list of states including 'Pendiente', 'Pdte. valoración', 'Pdte. presupuesto', 'Pdte. aceptación', 'Aceptado', 'En curso', 'En curso (Aplazado)', 'En curso (Pdte. cliente)', 'En validación DEV', 'En validación TEST', 'En validación PRO', 'Pdte. instalación'). There are also checkboxes for 'Planificar' (checked) and 'F.planificación' (checked).

The main area has two tables:

PROYECTO	PROCESO	NOMBRE	FACT.	HORAS PREV.	HORAS REALIZ.	HORAS VALIDA.	HORAS PTES.	ESTADO	FECHA PLANIFICACIÓN
SEMANA EN CURSO : 14 - 2024			Horas Pendientes: 16.5	Horas Validación: 0					
7000 - VEHICULOS 00.14.2024.01	MÓDULO SEGURIDAD USUARIOS (AFNORD)	No	60	43.5		16.5		En curso	03/04/2024

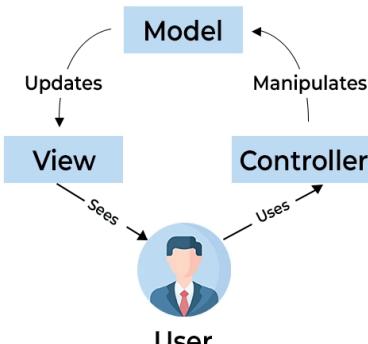
FECHA	Nº PARTES	HORAS FACT.	HORAS NOFACT.	HORAS TOTAL
03/04/2024	3	0	8.5	8.5
02/04/2024	2	0	8.5	8.5
27/03/2024	4	0	8.5	8.5
26/03/2024	4	0	8.5	8.5
25/03/2024	4	0	8.5	8.5
22/03/2024	3	0	6	6
21/03/2024	3	0	8.5	8.5
20/03/2024	3	0	8.5	8.5

At the bottom, it says 'TOTAL MES - H.FACT: 0 H.NOFACT: 17 H.TOTAL: 17'.

Partes10

5. Tecnologías usadas

Características de Laravel:



MVC



ORM Eloquent



Blade Template



Enrutamiento



Seguridad integrada

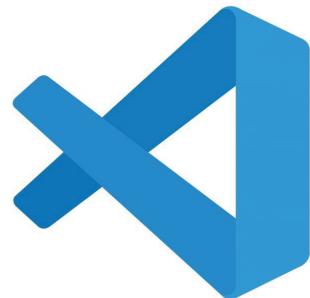
5. Tecnologías usadas

Lenguajes utilizados:



5. Tecnologías usadas

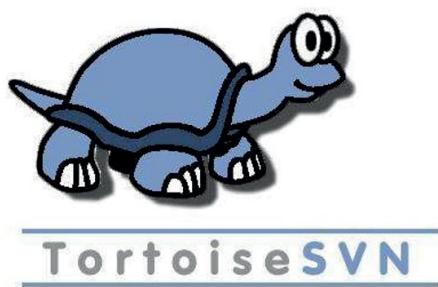
Aplicaciones utilizadas:



Visual Studio



Xampp



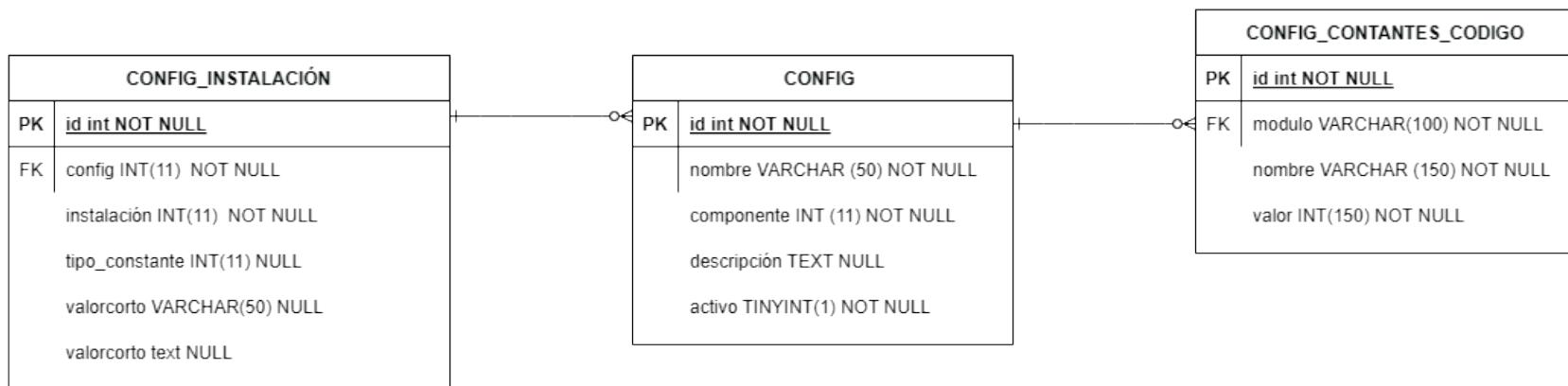
TortoiseSVN



ChatGPT

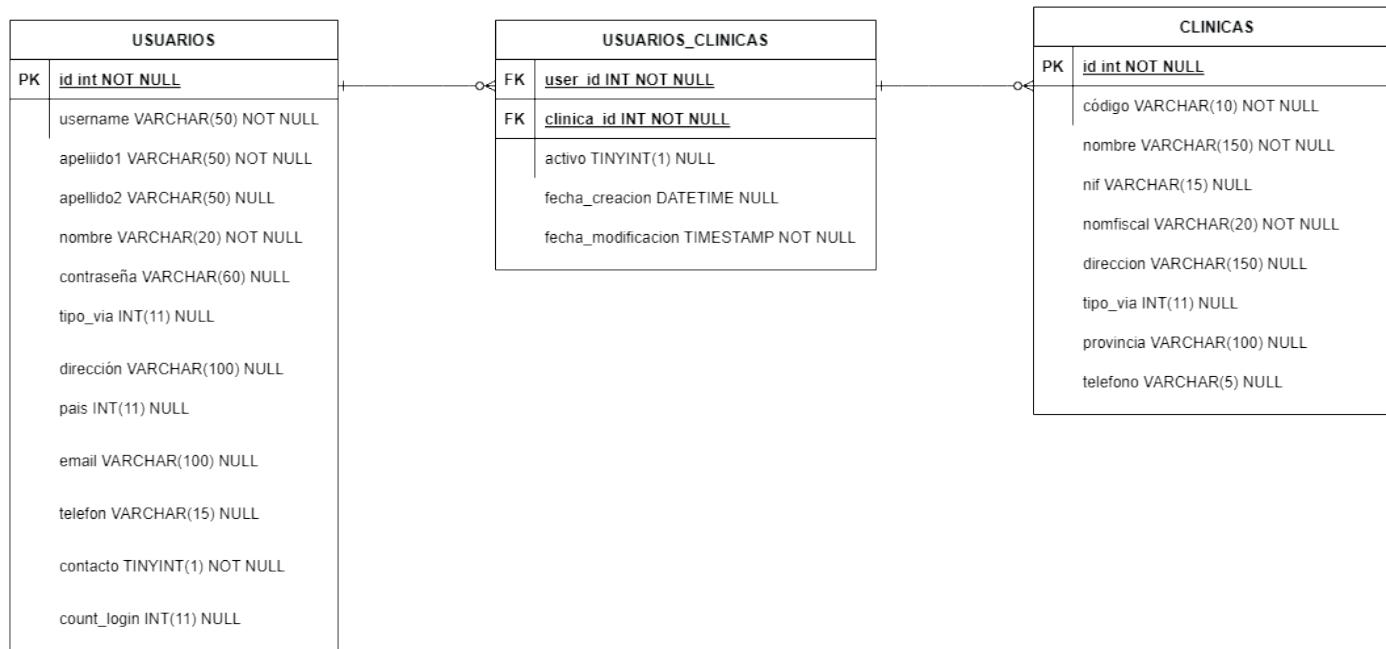
6. Base de datos

Relación config:



6. Base de datos

Relación usuarios:



7. Back end

Caducidad de contraseñas:

```
class CaducidadController extends Controller
{
    public function __construct()
    {
        $this->middleware('guest');
    }

    public function caducidadContrasenya()
    {
        ...
    }
}
```

7. Back end

Validación de contraseñas al crear contraseñas:

```
/**  
 * Show the form for creating a new resource.  
 *  
 * @return Response  
 */  
public function create() { ...  
}
```

```
function password_validator(password) {  
    var reglasJson = {!! json_encode($reglas) !!};  
  
    var configs = {!! json_encode($configs) !!};  
  
    if (reglasJson.longitud && reglasJson.mayuscula && reglasJson.minuscula && reglasJson.digito && reglasJson.especial) {  
        const regex = new RegExp(  
            `^(?=.*[a-z])` + // al menos una minúscula  
            `(?=.*[A-Z])(${reglasJson.mayuscula})` + // al menos dos mayúsculas  
            `${reglasJson.minuscula} > 0 ? "(?=.*[a-z])" : ""` + // condición para minúsculas si es necesario  
            `${reglasJson.digito} > 0 ? "(?=.*[\d])" : ""` + // condición para dígitos si es necesario  
            `${reglasJson.especial} > 0 ? "(?=.*[@$!%*?&])" : ""` + // condición para no alfanuméricos si es necesario  
            `[a-zA-Z\\@$!%*?&]+`  
        );  
        if (regex.test(password)) {  
            return true;  
        }  
    } else {  
        return false;  
    }  
}
```

```
else if (configs === '1' && password && !password_validator(password)) {  
    toastr.warning(`La contraseña debe tener al menos ${reglas.longitud} caracteres, incluir al menos ${reglas.mayuscula} mayúscula, ${reglas.minuscula} minúscula, ${reglas.digito} número y ${reglas.especial} carácter especial.`, '¡Atención!');  
    return;  
}
```

7. Back end

Validación de contraseñas al editar un usuario:

```
/**  
 * Show the form for editing the specified resource.  
 *  
 * @param int $id  
 * @return Response  
 */  
public function edit($id, Request $request) { ...  
}
```

```
/**  
 * Update the specified resource in storage.  
 *  
 * @param int $id  
 * @return Response  
 */  
public function update($id, EditUserRequest $request) { ...  
}
```

```
else if(configs === '1'){

    if (!password_Validator(pass)) {
        toastr.warning(
            `La contraseña debe tener al menos caracteres ${reglas.longitud}, incluir al menos ${reglas.mayuscula} mayúscula, ${reglas.minuscula} minúscula, ${reglas.digito} número y ${reglas.especia} carácter especial.`,
            '|Atención!');
        return;
    }
    else if(!pass){
        toastr.warning('Por favor, debe elegir una contraseña.', '|Atención!');
    }
}
```

7. Back end

has() -> any()

```
@if($errors->has())
    <div class="alert alert-warning" role="alert">
        @foreach ($errors->all() as $error)
            <div>{{ $error }}</div>
        @endforeach
    </div>
@endif
```

```
@if($errors->any())
    <div class="alert alert-warning" role="alert">
        @foreach ($errors->all() as $error)
            <div>{{ $error }}</div>
        @endforeach
    </div>
@endif
```

7. Back end

list() -> pluck()

```
$data->nombre_caja = $caja->list('nombre');
```

```
$data->nombre_caja = $caja->pluck('nombre');
```

7. Back end

Archivo de rutas:

```
Route::group(['prefix' => '', 'namespace' => 'Admin'], function() {
    Route::resource('tiposClientes', 'TiposClientesController');
    Route::match(['post', 'get'], 'tiposClientes', [ 'as' => 'tiposClientes.index', 'uses' => 'TiposClientesController@index']);
    Route::match(['post', 'get'], 'tiposClientes/create', [ 'as' => 'tiposClientes.create', 'uses' => 'TiposClientesController@create']);
    Route::match(['post', 'get'], 'tiposClientes/{id}/edit', [ 'as' => 'tiposClientes.edit', 'uses' => 'TiposClientesController@edit']);
    Route::match(['post', 'get'], 'tiposClientes/store', [ 'as' => 'tiposClientes.store', 'uses' => 'TiposClientesController@store']);
});
```

```
Route::group(['prefix' => '', 'namespace' => 'Admin'], function () {
    Route::resource('tiposClientes', 'TiposClientesController');
    Route::match(['post', 'get'], 'tiposClientes', [TiposClientesController::class, 'index'])->name('tiposClientes.index');
    Route::match(['post', 'get'], 'tiposClientes/create', [TiposClientesController::class, 'create'])->name('tiposClientes.create');
    Route::match(['post', 'get'], 'tiposClientes/{id}/edit', [TiposClientesController::class, 'edit'])->name('tiposClientes.edit');
    Route::match(['post', 'get'], 'tiposClientes/store', [TiposClientesController::class, 'store'])->name('tiposClientes.store');
});
```

7. Back end

count() -> count(array):

```
if(count($mapasEspecialidades) || count($mapasImgPruebas) || count($mapasVisitas)){
    $arrayReturn['borrado'] = 'NO';
    return $arrayReturn;
} else{
    $arrayReturn['borrado'] = 'OK';
    return $arrayReturn;
}
```

```
if(count($mapasEspecialidades) || count($mapasImgPruebas) || count($mapasVisitas)){
    $arrayReturn['borrado'] = 'NO';
    return $arrayReturn;
} else{
    $arrayReturn['borrado'] = 'OK';
    return $arrayReturn;
}
```

7. Back end

EditRequest/CreateRequest -> Request:

```
/**  
 * Store a newly created resource in storage.  
 *  
 * @return Response  
 */  
public function store(CreateUserRequest $request) { ...  
}
```

```
/**  
 * Store a newly created resource in storage.  
 *  
 * @return Response  
 */  
public function store(Request $request)  
{ ...  
}
```

```
/**  
 * Update the specified resource in storage.  
 *  
 * @param int $id  
 * @return Response  
 */  
public function update($id, EditUserRequest $request) { ...  
}
```

```
/**  
 * Update the specified resource in storage.  
 *  
 * @param int $id  
 * @return Response  
 */  
public function update($id, Request $request)  
{ ...  
}
```

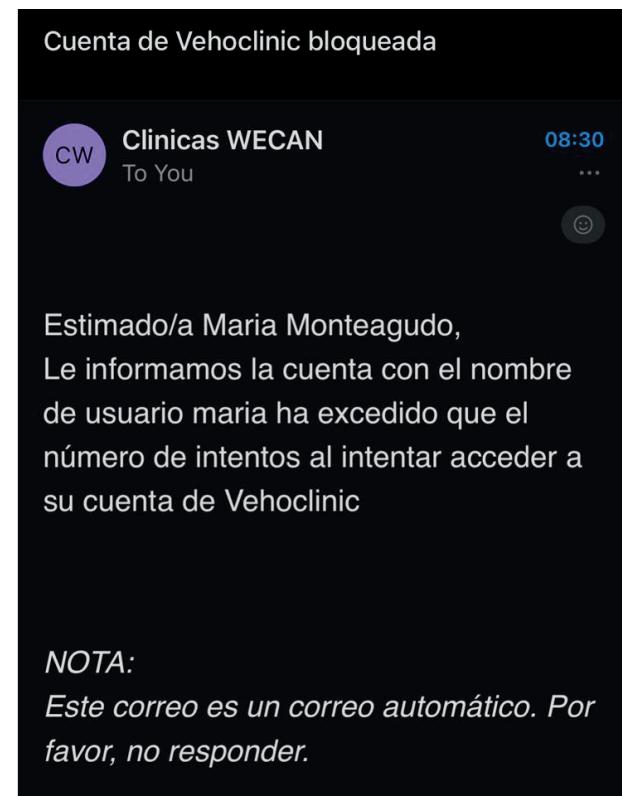
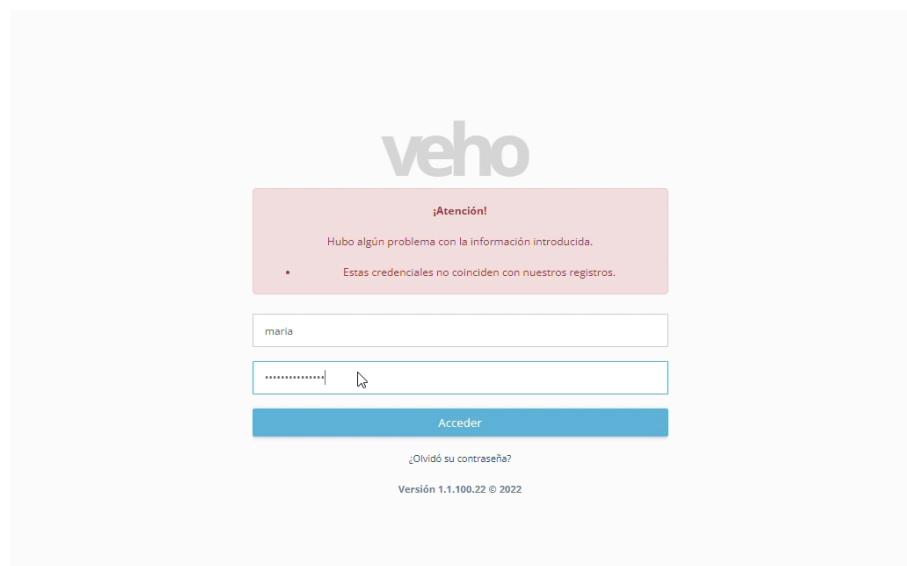
8. Problemas y dificultades

Los principales problemas han sido:

- Inexperiencia.
- Extensa estructura de tablas y relaciones.
- Hilos
- Librerías obsoletas (Filepicker y withdraw)

9. Resultado obtenido

Bloqueo usuario intento inicio sesión



9. Resultado obtenido

Contraseña al crear y edición de usuario

The screenshot shows a user creation/editation form. In the 'Contraseña' (Password) field, the user has entered 'mamoro02'. A tooltip message in the top right corner states: 'Atención La contraseña debe tener al menos caracteres 8, incluir al menos 1 mayúscula, 1 minúscula, 1 número y un carácter especial.'

Datos personales:
Nombre: María
Primer apellido: Montagudo
Segundo apellido: Robledillo
NIF: 84304151C
Usuario: mamoro02
Contraseña: mamoro02

Dirección:
Localidad: Solo del centro
Municipio:
Provincia:
País:

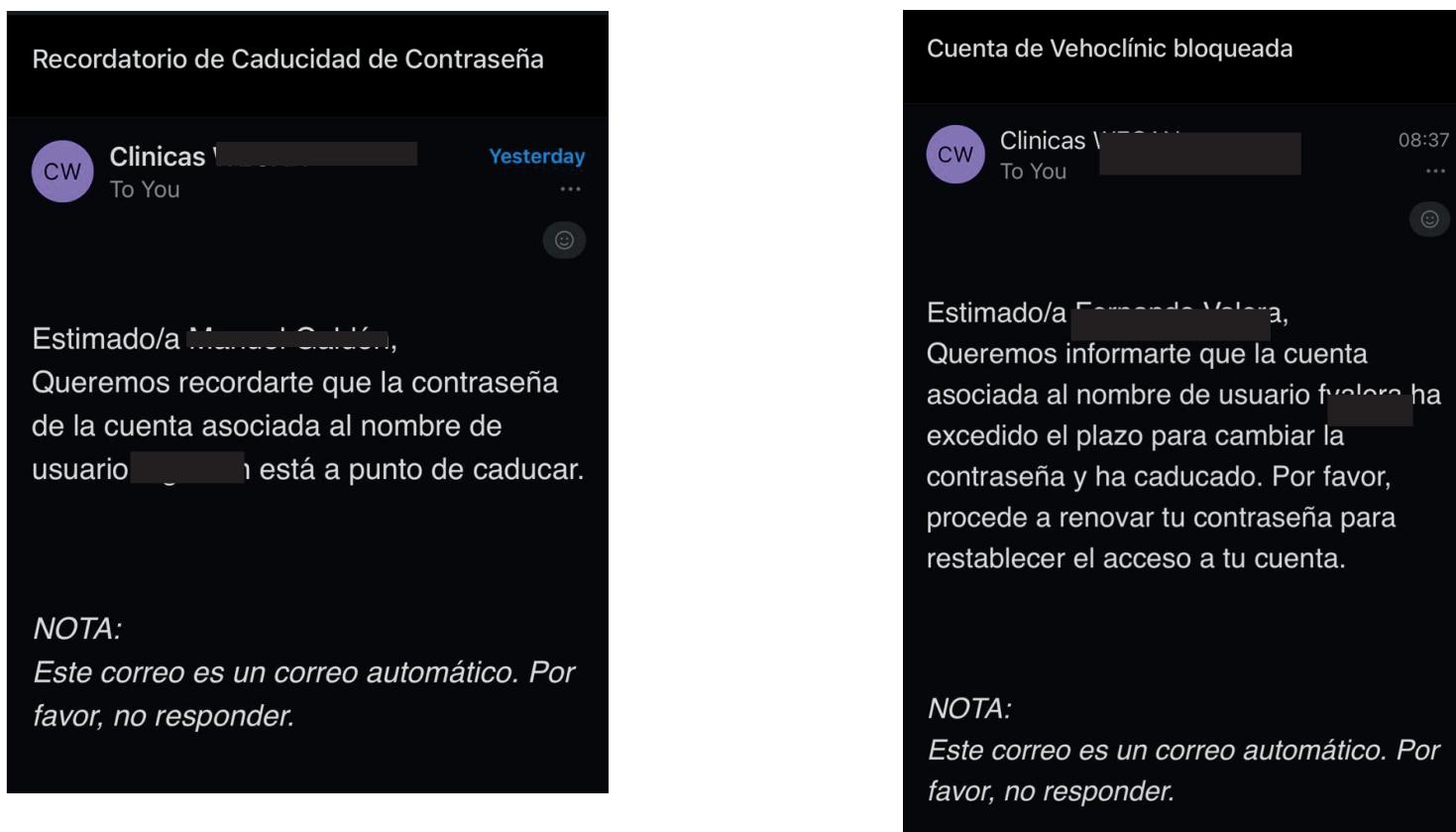
Datos adicionales:
Teléfono 1: Introduce número de teléfono
Teléfono 2: Introduce número de teléfono
E-mail: Introduce e-mail
Perfiles: Administrador del sistema
Centro: SECRETARIA CENTRAL
Centro preferente: SECRETARIA CENTRAL
Agenda preferente para SECRCENT: SECRET
Vesinario para recetas: Activado
Personas de contacto: Realizado curso de formación: Permisos propios
Categoría profesional: Vendedor: Interno

Botones: Guardar, Cancelar

```
{"validez": "90", "recordatorio": "10", "longitud": "9", "mayuscula": "1", "minuscula": "1", "digito": "1", "especial": "1", "repeticion": "365", "bloqueo": "3"}
```

9. Resultado obtenido

Crontab



10. Lineas futuras

Los principales problemas han sido:

- Firma electrónica en una tablet.
- Mejora en el diseño responsive.
- Depurar y optimizar el código.
- Integración de otros proveedores en la aplicación.
- Mejorar las consultas SQL utilizadas en la aplicación

11. Conclusión



12. Webgrafía

- Documentación Laravel 5
- Documentación Laravel 10
- Stack Overflow
- ChatGPT