
This is the **published version** of the bachelor thesis:

Ginfarré Blázquez, Esther; Figuerola Torrell, Adrià, dir. Optimització i modernització del sistema de gestió de Ralarsa Tallers : disseny, desenvolupament i documentació. 2024. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/290101>

under the terms of the  license

Optimització i Modernització del Sistema de Gestió de Ralarsa Tallers: Disseny, Desenvolupament i Documentació

Esther Ginfarré Blázquez

8 de febrer de 2024

Resum– Aquest projecte se centra en el desenvolupament i implementació d'un nou sistema de gestió per a tallers de reparació i substitució de vidres d'automòbils per a l'empresa Ralarsa, utilitzant tecnologies avançades com Microsoft Graph, ASP.NET i Angular. L'objectiu principal és millorar l'eficiència operativa i augmentar la seguretat en les operacions dels tallers, superant els desafiaments de l'entorn de gestió actual a través de l'aplicació d'una metodologia de desenvolupament en espiral. Els resultats esperats inclouen una major eficiència en la gestió dels processos de taller, una seguretat de dades millorada i una major facilitat d'ús per als usuaris finals.

Paraules clau– ASP.NET, Angular, Microsoft Graph, eficiència operativa, gestió de tallers, metodologia en espiral, seguretat de dades.

Abstract– This project focuses on the design and implementation of a new management system for automotive glass repair and replacement workshops for the company Ralarsa, utilizing advanced technologies such as Microsoft Graph, ASP.NET, and Angular. The main objective is to enhance operational efficiency and increase security in workshop operations, overcoming the challenges presented by the current management environment through the application of a spiral development methodology. Expected outcomes include improved efficiency in workshop process management, enhanced data security, and increased usability for end-users.

Keywords– ASP.NET, Angular, Microsoft Graph, operational efficiency, workshop management, spiral methodology, data security.

1 INTRODUCCIÓ

AMB l'avanç continu de la tecnologia, l'eficiència i la seguretat han esdevingut elements crucials en la gestió empresarial. En aquest context, Ralarsa[1], una empresa amb una trajectòria de més de quatre dècades en el sector de la reparació i substitució de vidres d'automòbils, s'enfronta al repte de renovar el seu sistema de gestió de tallers. Com a pionera en la creació de la primera xarxa de tallers especialitzats en cristalleria d'automòbils, Ralarsa continua compromesa amb la millora contínua i la

innovació. Aquesta necessitat sorgeix en un context en què la tecnologia específica al sector del vidre d'automòbil ha avançat significativament, posant en relleu les limitacions de l'entorn de gestió actual de l'empresa.

El programa actual de gestió del taller presenta desafiaments significatius, ja que té més de 50 arxius PHP i utilitza llibreries obsoletes. Es va crear aquest programa quan l'empresa disposava de menys tallers que gestionar, pel qual no es va estructurar de forma adequada des del començament. Amb el temps, a mesura que l'empresa creixia, s'han acumulat pegats i solucions temporals per abordar els problemes emergents. Ara, l'empresa està buscant millorar la seva eficiència i seguretat, i a causa de la recent compra de Cary Group [2], es reconeix la importància tecnològica i el control, la qual cosa possiblement podria portar a una migració pròxima a un sistema CRM o ERP [3].

Ens trobem que el principal desafiament que enfronta l'empresa és que el seu entorn de gestió actual és inefici-

• E-mail de contacte: 1494785@uab.cat
• Menció realitzada: Tecnologies de la Informació
• Treball tutoritzat per: Adrià Figuerola (Àrea de Ciències de la Computació i Intel·ligència Artificial)
• Curs 2023/24

ent i a més a més, és potencialment insegur. Ens trobem, també, que el codi i les estructures actuals són extensos, sense comentaris i desorganitzats, la qual cosa dificulta el manteniment i les posteriors millores.

Per tant, la necessitat és millorar la gestió dels tallers, per augmentar-hi l'eficiència i la seguretat. Per incorporar aquesta millora en seguretat, es planteja utilitzar Microsoft Graph [4], un programa que serà més fàcil d'implementar pel fet que l'empresa, avui en dia, fa ús de l'Office 365 en els sistemes d'autenticació.

Aquest projecte de Treball de Fi de Grau té com a objectiu principal dissenyar, desenvolupar i documentar un nou entorn de gestió per als tallers de Ralarsa. Aquest nou entorn no només optimitzarà els processos existents, sinó que també incorporarà millores significatives en seguretat tal com s'ha comentat en els punts anteriors.

L'estructura general d'aquest article serà la següent. En primer lloc, a la secció 2 definirem els objectius que volem aconseguir. En segon lloc, a la secció 3 descriurem la metodologia utilitzada en aquest projecte. En tercer lloc, a la secció 4 plantejarem el cronograma del projecte, establint les tasques principals i els seus terminis. A la secció 5 exposarem l'estat de l'art dels diferents aspectes d'aquest projecte, amb un enfocament en l'adaptabilitat i la flexibilitat requerides per afrontar els canvis que puguin sorgir. A la secció 6 s'explicarà parts del desenvolupament i exemples de la seva implementació. Per continuar, a la secció 7 es farà una breu discussió dels resultats. I finalment a la secció 8 podrem trobar les conclusions i per últim, a la secció 9 els agraïments.

2 OBJECTIUS DEL PROJECTE

El principal propòsit d'aquest projecte és desenvolupar un sistema de gestió integral per a Ralarsa, enfocat a millorar l'eficiència operativa i augmentar la seguretat en les operacions dels tallers. Aquest sistema, que incorporarà Microsoft Graph de l'Office 365 com a eina central, pretén superar els desafiaments d'ineficiència i manca de seguretat en l'entorn actual de gestió dels tallers. Amb aquesta iniciativa, busquem proporcionar una solució innovadora i eficaç que no només optimitzi els processos ja existents, sinó que també garanteixi la protecció i l'accés segur a dades crítiques.

Donat que aquest és un projecte d'ampli abast, en aquesta etapa inicial, ens concentrarem en el disseny, desenvolupament de la pàgina principal i la gestió de les fulles de treball. Això inclourà aspectes com l'anàlisi i la implementació del cercador, el llistat de fulles de treball i la capçalera. També abordarem les funcionalitats d'alta, baixa i modificació de les fulles de treball. Amb aquest enfocament, estem compromesos a proporcionar una solució integral que satisfarà les necessitats, immediates, de millora en la gestió dels tallers de Ralarsa.

Es dividirà el projecte en els següents objectius específics per guiar-nos al llarg d'aquest, i això permetrà centrar els esforços en les àrees més rellevants per assolir l'objectiu principal:

- **Anàlisi del sistema actual i descripció dels processos necessaris:** Es realitzarà una revisió exhaustiva del sistema existent, identificant les àrees de millora i descri-

vint els processos i condicions actuals, per tal de disposar d'una guia per les posteriors actualitzacions.

- **Conservació de funcionalitats actuals del programa:** Cal assegurar que tots els programes existents continuïn funcionant com fins ara, d'aquesta manera només s'hauran de fer petites millores, per aconseguir un bon funcionament de la infraestructura.
- **Modernització i centralització de sistemes:** S'actualitzaran i unificaran els sistemes per garantir una gestió més eficient i eficaç de la informació i dels recursos.
- **Autenticació centralitzada utilitzant Office 365:** Implementar una autenticació de doble factor, homogeneïtzada, utilitzant les credencials d'Office 365 per garantir la seguretat de l'accés al sistema.
- **Optimització de les crides al servidor:** Millorar l'eficiència de les crides al servidor per reduir el temps de resposta i augmentar la velocitat del sistema.
- **Fàcil adaptació per a una posterior migració a ERP:** Dissenyar el sistema de manera que sigui fàcilment adaptable per una futura migració cap a un sistema ERP si és necessari.
- **Creació de l'estructura de Ralarsa Tallers:** Desenvolupar una base sòlida que permeti l'escalabilitat i ampliació del sistema sense comprometre la seva centralització i funcionalitat.

3 METODOLOGIA DE DESENVOLUPAMENT EN ESPIRAL

Proposada inicialment per Barry Boehm en 1988 a través del seu article "A Spiral Model of Software Development and Enhancement" [5], la metodologia de desenvolupament en espiral representa una alternativa innovadora als models tradicionals de desenvolupament de programari, com ara el model en cascada. Aquest enfocament és caracteritzat per la seva naturalesa iterativa i incremental, oferint una major flexibilitat i adaptabilitat en comparació amb els mètodes convencionals.

Hem escollit aquest model per diverses raons clau: [6]:

1. **Complexitat del Projecte:** La metodologia en espiral és idònia per projectes complexos, com el nostre, donant-nos l'agilitat per adaptar-nos als canvis i requisits emergents.
2. **Enfocament iteratiu:** És crucial identificar i abordar els problemes de forma progressiva, evitant així complicacions majors en etapes avançades.
3. **Flexibilitat i Adaptabilitat:** La possible integració futura amb sistemes com CRM o ERP requereix una metodologia que permeti aquesta evolució sense grans modificacions.
4. **Millora Contínua:** Alineada amb la filosofia de Ralarsa, d'innovació constant, la metodologia en espiral facilita l'actualització i millora continuada del sistema.

3.1 Tipus d'Espirai

Per aconseguir l'objectiu principal, implementarem una metodologia estructurada en cinc iteracions:

- **Iteració 1 (Avaluació i Planificació):** Definirem els problemes i necessitats del sistema actual, establint un pla d'acció clar.
- **Iteració 2 (Anàlisi de Requisits):** Documentarem en detall totes les funcionalitats i característiques del sistema actual, creant una base sòlida per a futures millores.
- **Iteració 3 (Disseny del Sistema):** Desenvoluparem una nova arquitectura per al sistema, incloent-hi el disseny del back-end, basat en l'anàlisi elaborada.
- **Iteració 4 (Implementació i Codificació):** Traduirem el disseny en codi funcional, respectant les especificacions establertes.
- **Iteració 5 (Proves i Avaluació):** Testejarem exhaustivament el sistema per assegurar-ne la qualitat i l'eficàcia, realitzant les modificacions necessàries.

En l'adopció de la metodologia de desenvolupament en espiral per a aquest projecte, hem incorporat elements clau com la planificació flexible i una gestió de riscos proactiva. Això ens ha permès adaptar-nos amb eficàcia als canvis imprevistos i gestionar de manera més eficient els recursos del projecte. A més, s'han mantingut reunions setmanals amb el tutor per revisar el progrés, ajustar plans segons sigui necessari i garantir que el projecte continuï alineat amb els objectius establerts.

Aquest enfocament integrat dins de la metodologia utilitzada ha estat fonamental per afrontar amb èxit les complexitats del projecte. Ens ha permès no només complir amb els objectius establerts, sinó també manejar els recursos de manera més eficaç, assegurant la finalització exitosa de totes les fases planificades, incloent-hi aquells aspectes que requereixen una atenció especial.

4 PLANIFICACIÓ AVANÇADA DEL PROJECTE

A la primera etapa del projecte, hem definit les tasques principals a realitzar, caracteritzant-nos per la divisió de grans tasques en subtasques més específiques i detallades. Hem triat fer ús de Microsoft Projects per a una gestió eficaç. Això ens permet, no només un seguiment rigorós de l'avanç del projecte, sinó que també una definició precisa i detallada de cada tasca.

En la figura 1 es presenta un cronograma detallat del projecte, on es destaquen les principals tasques que formaran part de les iteracions 2, 3 i 4. A continuació, es detallaran les més significatives:

- **Disseny (front-end):** Sota la direcció del dissenyador de l'empresa, es durà a terme una renovació de les interfícies d'usuari actuals, amb un focus especial en la millora de la pàgina de gestió de taller.
- **Anàlisi i disseny (back-end):** Aquesta tasca serà dividida en dues parts fonamentals:

- Elaboració d'un *Diagrama de Funcionament*, el qual descriurà les interaccions amb la base de dades i la lògica interna de l'aplicació.
- Creació d'un *Diagrama de Flux*, enfocat a les vistes que es desenvoluparan.

Avançarem en el projecte dividint les tasques en dues categories principals: Back-End i Front-End. Dins de cada categoria, es detallaran tasques específiques, que permetran un desenvolupament eficient i ordenat d'aquesta estructura organitzada.

• Back-End:

- *Desenvolupament de la Pàgina Principal:* Inclou la gestió d'e-mails no llegits, visualització d'import d'albarans pendents i el seu número associat.
- *Desenvolupament del Buscador de Fulles:* Creació d'un sistema de cerca per a clients, amb funcionalitats de càrrega de dades i filtres.
- *Llistat de Fulles:* Implementació de controls per al canvi d'estat de les fulles.
- *Creació/Modificació de Fulles:* Gestió de la creació i edició de fulles, incloent-hi el control de càrrega de dades dels selectors específics.
- *Detall d'Articles:* Desenvolupament de la funcionalitat per a la gestió d'articles en les fulles, incloent-hi inserció, filtres i tarifació.

• Front-End:

- *Pàgina Principal:* Disseny del menú de navegació i de la capçalera, juntament, amb la selecció de taller i de l'idioma.
- *Buscador de Fulles:* Desenvolupament de la interacció i gestió del formulari amb els seus corresponents filtres.
- *Llistat de Fulles:* Implementació del llistat i la seva paginació.
- *Creació/Modificació de Fulles:* Gestió de la interacció dels formularis, com dels filtres associats d'aquests.
- *Detall d'Articles:* Desenvolupament de la interacció, gestió dels formularis, creació de les taules d'articles i dels seus corresponents filtres.

Utilitzarem el Planner de Microsoft Office [7] juntament amb Microsoft Projects per fer un seguiment detallat de les tasques. Això inclourà actualitzacions diàries en el Planner per part del desenvolupador i revisions periòdiques en les reunions setmanals amb el gestor de projectes, on es registraran els progressos en el Projects.

5 ESTAT DE L'ART

L'anàlisi de l'estat de l'art d'aquest projecte es focalitza en els seus components clau. Això inclou la transició de PHP a tecnologies més avançades, com Angular per al front-end i ASP.NET per al back-end. A més, s'explora la integració amb Microsoft Graph, una decisió clau que reflecteix les tendències actuals en desenvolupament web i les millors pràctiques de la indústria.

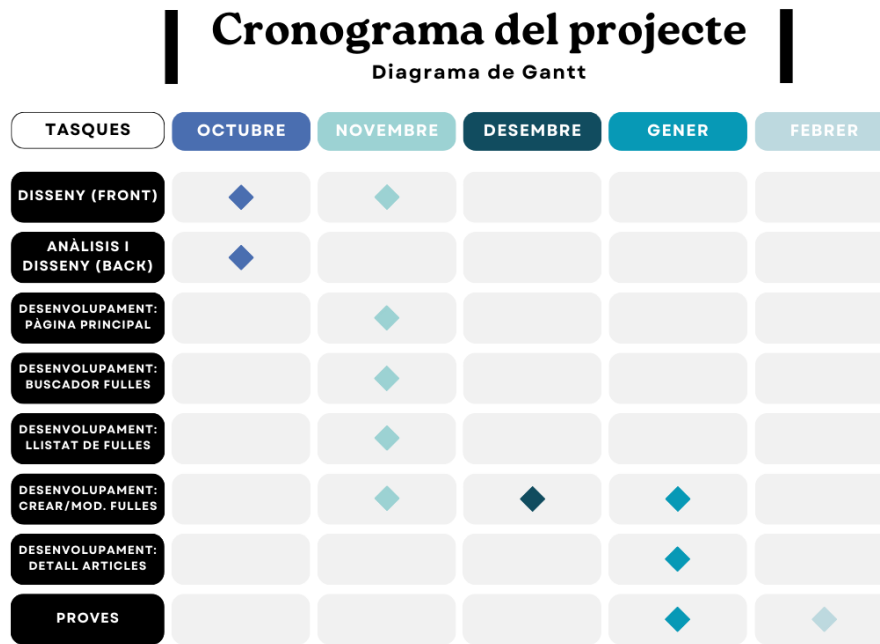


Fig. 1: Diagrama de Gantt

5.1 Migració de PHP a Angular i ASP.NET

- Angular [8]: Aquest framework de JavaScript basat en TypeScript, desenvolupat per Google, és reconegut per la seva eficàcia en la creació d'aplicacions web dinàmiques. La seva capacitat per a oferir vinculació de dades bidireccional, injecció de dependències, i una arquitectura modular, el converteix en una eina ideal per afrontar els desafiaments del front-end.
- ASP.NET [9]: Part de la família Microsoft, aquesta plataforma destaca per la seva robustesa, seguretat i escalabilitat, essent una elecció òptima per al desenvolupament del back-end. ASP.NET facilita la integració amb altres productes de Microsoft i suporta múltiples patrons de disseny.
- Microsoft Graph [10] [11]: Aquesta API permet una integració sofisticada amb diversos serveis de Microsoft 365, com correus electrònics, calendaris i documents d'Office, potenciant així la funcionalitat i eficiència del projecte.

escalables. Aquests microserveis s'han creat de forma interna a través d'ASP.NET i fent la implementació del "controller" o "api" que cridarà al client Angular.

Els microserveis han sigut creats a partir del llenguatge GraphQL. En la figura 2 es mostra un diagrama abstracte del funcionament de l'arquitectura centrat a l'inici de sessió i la primera càrrega d'informació, en aquest cas dels tallers del desplegable i la informació que s'ha de validar per donar accés o no a l'usuari a les funcionalitats.

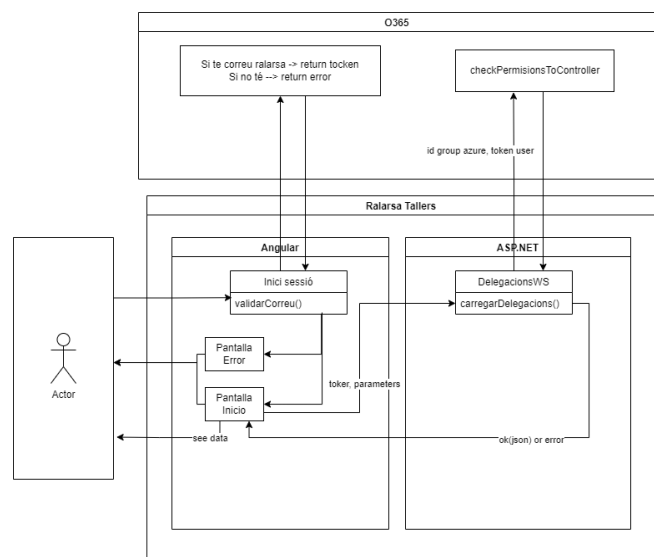


Fig. 2: Arquitectura abstracta de les interaccions de les diferents parts de l'aplicació

6 DESENVOLUPAMENT

Després d'una anàlisi detallada realitzada durant la segona iteració, hem centrat els nostres esforços en el desenvolupament i implementació de consultes dins de l'entorn ASP.NET. A més, hem treballat en la crida i estilització de components utilitzant Angular, seguint un model basat en GraphQL [12]. Aquest enfocament permet als usuaris sol·licitar exactament el que necessiten, optimitzant així l'eficiència i la precisió del sistema.

6.1 Arquitectura

L'arquitectura utilitzada en el projecte es basa en els microserveis, creant dissenys d'aplicacions flexibles i altament

L'exemple del diagrama de l'arquitectura se centra en els components que es veuran en les posteriors seccions.

6.2 Implementació amb ASP.NET

6.2.1 Distribució fitxers

La configuració del projecte ASP.NET ha estat meticulosament dissenyada per a assegurar una organització lògica i eficient del codi. Centrem la vista en el controlador a utilitzar per fer la càrrega principal dels tallers.

Com es pot veure a la figura 3, la nostra estructura de carpetes és intuïtiva, facilitant l'accés ràpid i l'actualització dels arxius.

Dins de la carpeta principal "Clases", hem ubicat les classes que representen directament les taules de la nostra base de dades. Cada classe està definida amb els seus atributs corresponents, detallats en la figura 5, i les seves funcions específiques, il·lustrades a la figura 4. Aquesta disposició no només proporciona una visió clara de la nostra estructura de dades, sinó que també facilita el manteniment i l'escalabilitat del sistema, ja que s'ha fet ús de GraphQL.

Cadascuna de les classes, estan estructurades de manera que contenen tota la informació necessària per interactuar amb la seva taula corresponent en la base de dades. Això inclou definicions d'atributs que interconnecten directament amb els camps de la base de dades, així com mètodes per a la manipulació de dades (com consultes, actualitzacions i insercions).

Aquest enfocament modular no només optimitza el desenvolupament, sinó que també permet als desenvolupadors comprendre ràpidament l'estructura i la lògica del projecte, facilitant així la col·laboració i les millores posteriors.

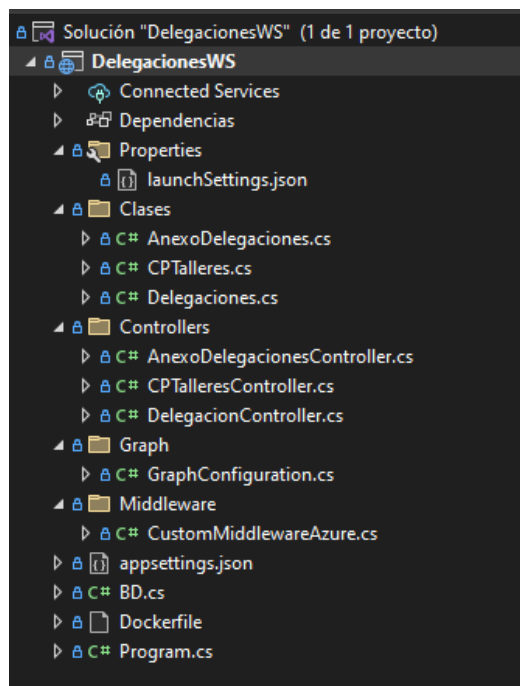


Fig. 3: Estructura arxius controlador Delegacions

Els fitxers de la carpeta "Controllers" facilita la interacció del front-end (aplicació client) amb la base de dades per efectuar operacions necessàries relacionades amb la classe, i controlar les respostes i els errors d'aquestes.

6.2.2 Integració amb Graph i grups Azure

Una part crucial del nostre desenvolupament ha estat la integració eficaç amb Microsoft Graph i Azure, dos components clau de l'ecosistema de Microsoft. Per a això, hem implementat un sistema de grups d'accés específics per a cada controlador dins de l'aplicació. Aquests grups permeten gestionar de manera final els permisos d'usuari, assegurant que només els usuaris autoritzats puguin accedir a funcions específiques.

Per implementar aquest sistema, hem utilitzat el Centre d'Administració de Microsoft 365, on es creen i gestionen aquests grups d'accés. Cada vegada que desenvolupem un nou controlador, establim un grup d'accés associat a ell. Això ens permet assignar un identificador únic de Graph (graph id) a cada grup, el qual és essencial per la correcta autenticació i autorització d'usuaris dins del nostre sistema. Després de crear aquest nou grup s'ha d'associar els membres que tenen accés i els quals podran interactuar amb el controlador.

Aquesta configuració no només enforteix la seguretat, sinó que també proporciona una gran flexibilitat. Podem controlar fàcilment qui té accés a quines parts de l'aplicació, i això ens permet adaptar l'aplicació a les necessitats canviants de l'empresa i als diferents rols d'usuari. A més, aquesta integració amb Graph i Azure facilita la futura expansió i escalabilitat del sistema, permetent-nos afegir noves funcionalitats i controladors amb facilitat i seguretat.

Aquestes inicialitzacions de paràmetres es fan a "Graph-Configuration.cs" que s'observa a la figura 3, per després utilitzar-les a "Program.cs", indicant així tot el necessari que ha de tenir la crida al controlador com el tipus d'autenticació a utilitzar.

6.2.3 Modularització i Reutilització

Un aspecte fonamental en la nostra implementació amb ASP.NET ha estat la creació d'una llibreria específica per a la modularització i reutilització del codi. Aquesta tasca s'ha dut a terme en estreta col·laboració amb el tutor del projecte, combinant la seva experiència amb les nostres necessitats específiques. La llibreria desenvolupada facilita la realització de crides a la base de dades de manera segura i eficient, permetent passar paràmetres de forma controlada i segura.

Aquesta aproximació modular ha estat clau per millorar la nostra eficiència de desenvolupament. Amb aquesta llibreria, les funcions comunes i repetitives se centralitzen, reduint així la redundància i augmentant la coherència en tot el projecte. A més, la reutilització de components ha facilitat l'escalabilitat i la mantenibilitat de l'aplicació, permetent-nos expandir i adaptar el sistema amb menys esforç i major confiança.

La llibreria inclou funcionalitats per a la gestió de consultes, manipulació de dades, i tractament d'errors, entre altres. Això no només simplifica el procés de desenvolupament, sinó que també millora la qualitat del codi, facilitant la seva mantenibilitat, seguretat i llegibilitat. La modularització també promou una major separació de preocupacions, on cada mòdul o component s'encarrega d'una funcionalitat específica, facilitant així la depuració i les proves del codi.

Com es pot observar al fragment de codi de la figura 4 la consulta es forma a través de les crides de la llibreria "Pa-

rameters” per formar la consulta segons el JSON passat per l’aplicació “Client” que en aquest cas seria Ralarsa Tallers a través d’Angular. Això permet que cadascun dels controladors que es creen d’ASP.NET siguin més personalitzables.

```

1 referencia
public static List<Dictionary<string, object>> GetTabla(Parameters.Parameters param)
{
    Dictionary<string, string[]> Data = Parameters.Parameters.getRegexDictionary(new Delegaciones());
    string connstring = BD.getConexion();
    NpgsqlConnection conn = new NpgsqlConnection(connstring);
    conn.Open();
    NpgsqlCommand cmd = new NpgsqlCommand();
    cmd.Connection = conn;
    string filter = param.getFilters(ref cmd, Data);
    string sel = param.getSelect();
    string orden = param.getOrden(Data);
    string limit = param.getLimit();
    string offset = param.getOffset();
    // set the query and parameters of the command
    string query = "SELECT " + sel + " FROM " + TABLE_NAME + " " + filter + orden + limit + offset;
    cmd.CommandText = query;
    NpgsqlDataReader reader = cmd.ExecuteReader();
}

```

Fig. 4: Funció d’obtenció de dades al controlador DelegacionsWS: Creació de la consulta

A través de la crida de les funcions de les llibreries podem extreure la consulta i el filtre de la mateixa a realitzar, igual que l’ordre en el qual extreu les dades. Abans de realitzar tot això fem una crida a la funció *getRegexDictionary()* que a través de la declaració d’atributs crea el diccionari Regex que indica l’atribut, el tipus de dada i el regex corresponent. Això el que aconseguim és tenir un llistat dels atributs i els filtres a passar, per assegurar-nos de no carregar dades més grans de les permeses a la base de dades i de no permetre la injecció SQL.

A través de la classe *CustomProperties* de la llibreria “Parameters” que visualitzem als atributs de la classe en la figura 5, inicialitzem l’atribut segons el constructor per paràmetre. Aquest constructor accepta fins a tres paràmetres:

- String: Nom de l’atribut tal com és descrit a la base de dades
- Bool (optional): Indicador si es pot fer insert de l’atribut a la base de dades. Per defecte true.
- Bool (optional): Indicador si es pot fer update de l’atribut a la base de dades. Per defecte true.

```

10 referencias
public class Delegaciones
{
    public const string TABLE_NAME = "delegaciones";
    public const string DEFAULT_PREFIX = "d";

    [RegularExpression(@"^[0-9]+$", ErrorMessage = "Error: [Parameters.CustomProperties("delegacion", true)]
    2 referencias
    public int? delegacion { get; set; }

    [RegularExpression(@"^.{0,10}$", ErrorMessage = "Error: [Parameters.CustomProperties("nombre", true)]
    0 referencias
    public string? nombre { get; set; }
}

```

Fig. 5: Atributs de la classe Delegacions en el controlador DelegacionsWS

Aquelles dades que en fer un insert es generen automàticament com un *nextval()* d’un id o en disposar d’un

now() en un created s’hauran d’inicialitzar com a falsa, per no permetre l’insert.

Continuant amb la figura 4 ja només falta formar la consulta cridant al *getFilters*, *getSelet* i *getOrden*, *getlimit*, *getoffset*. A la primera d’aquestes li passem dos paràmetres, el primer és l’encarregat de crear la consulta per no permetre les injeccions SQL passant les dades per comandes. El segon paràmetre és el diccionari, que el que farà serà passar el filtre abans de crear la instància per la consulta. Si algun paràmetre és erroni o no passa el filtre donarà una excepció i es retornarà al client, en cas contrari continuarà la funció, i retornarà un JSON amb el resultat com s’observa a la figura 6.

```

// access the values of each column by using the name or index of the column
List<Dictionary<string, object>> resultado = new List<Dictionary<string, object>>();
while (reader.Read())
{
    Dictionary<string, object> fila = new Dictionary<string, object>();
    for (int i = 0; i < reader.FieldCount; i++)
    {
        string nombreCampo = reader.GetName(i);
        object valorCampo = reader.GetValue(i);
        fila.Add(nombreCampo, valorCampo);
    }
    resultado.Add(fila);
}

reader.Close();
conn.Close();

return resultado;
}

```

Fig. 6: Funció d’obtenció de dades al controlador DelegacionsWS: Extracció dades segons la consulta

En resum, la creació d’aquesta llibreria específica per a ASP.NET és un clar exemple de com la nostra estratègia de desenvolupament prioritza l’eficiència, la seguretat i la flexibilitat, preparant el terreny per a futures millores i expansions del sistema.

6.2.4 Implicacions del Desenvolupament

La nostra experiència amb la llibreria “Parameters” ha obert noves vies d’eficiència en el desenvolupament dins de l’entorn ASP.NET. Aquesta fase ha transcendit la simple organització de consultes; hem innovat en l’arquitectura de l’aplicació, establint un model de desenvolupament que emfatitza la flexibilitat i la reutilització de codi. A diferència de les etapes anteriors, on l’enfocament se centrava en l’estructuració i la integració, aquesta fase ha permès posar a prova l’eficàcia d’aquestes estratègies en escenaris reals d’aplicació.

El treball amb la llibreria no només ha estat una tasca de codificació; ha implicat una profunda reflexió sobre com estructurar un projecte per a maximitzar l’eficiència a llarg termini. Hem observat millores tangibles en la velocitat de desenvolupament i en la capacitat de resposta a canvis i nous requisits. Aquest enfocament, que combina la teoria amb la pràctica, ha enriquit el nostre coneixement col·lectiu i ha posat una base sòlida per a la innovació contínua.

En resum, aquesta etapa del projecte no només ha reforçat la nostra infraestructura tecnològica, sinó que també ha enriquit la nostra comprensió i aplicació de les millors pràctiques en el desenvolupament de programari. Aquestes experiències ens posen en una posició favorable per encarar els desafiaments futurs i explorar noves oportunitats d’innovació.

6.3 Implementació Angular

6.3.1 Creació d'interfície visual

En el procés de creació de la interfície visual, ens hem enfrontat a desafiaments significatius, especialment en l'equilibri entre la personalització i l'ús de plantilles predefinides. Per assolir aquesta fina balança, hem optat per utilitzar els components de Material Angular [13], una llibreria que ofereix un ventall de components d'interfície d'usuari reutilitzables i ben provats, basats en els principis de disseny de materials. Aquesta elecció ha accelerat el desenvolupament, però al mateix temps, ha presentat reptes únics en la personalització segons els requisits específics del nostre dissenyador.

La nostra estratègia s'ha centrat a adaptar aquests components estàndard perquè s'ajustin a la nostra visió estètica única i les necessitats funcionals. Aquesta tasca ha requerit una comprensió profunda tant dels components de Material Angular com dels nostres objectius de disseny. Hem aconseguit un equilibri entre utilitzar l'eficiència dels components predefinits i aplicar modificacions personalitzades per a reflectir la identitat visual de Ralarsa.

Aquest procés no només ha estat una qüestió de codificació, sinó també una oportunitat d'explorar com el disseny i la funcionalitat poden coexistir harmoniosament. El resultat ha estat una interfície d'usuari que no només és intuïtiva i atractiva visualment, sinó que també s'alinea amb les necessitats específiques dels usuaris finals del sistema.

Pel que respecta a les crides dins d'angular s'han fet ús d'interfícies per les crides de les funcions. Bàsicament, és la creació d'un objecte que es fa ús en el codi, per inicialitzar o crear els atributs html amb informació proporcionada per la base de dades o a l'invers.

6.3.2 Autenticació amb el Controlador

Un dels pilars de la nostra implementació Angular és la gestió eficient de l'accés dels usuaris als controladors basada en la seva autenticació mitjançant Office 365. Per aconseguir-ho, hem establert un sistema on cada controlador al servidor es defineix amb un àlies específic, facilitant així la seva identificació i accés.

El pas crític en aquesta interacció és l'ús de tokens d'autenticació. Cada usuari, quan intenta accedir a un controlador, ha de presentar un token únic obtingut a través del seu compte d'Office 365. Aquest token actua com una clau digital que verifica la identitat de l'usuari i els seus permisos. Quan un usuari realitza una crida a un controlador, el sistema primer verifica la validesa d'aquest token. Si el token és vàlid, la crida es processa i l'usuari obté accés a les dades o funcionalitats requerides. En cas contrari, l'accés és denegat, mantenint així la seguretat i la integritat de l'aplicació.

Per treure el token de l'usuari s'ha de fer ús de la llibreria MsalService. Aquesta llibreria, ens permet agafar informació de l'usuari que ha iniciat sessió gràcies als paràmetres de graph explorer, ja sigui coses com els missatges no llegits, com en aquest cas l'id del token que necessitem.

En la figura 7 es pot veure que es disposa d'un atribut anomenat msalService. Aquest atribut s'ha inicialitzat al constructor d'aquest component, amb l'scope "GroupMember.Read.All", es pot accedir al token de l'usuari com "resultado.accessToken".

```
ngOnInit() : void {
  this.msalService.acquireTokenSilent({ silentRequest: {
    scopes: ['GroupMember.Read.All']
  }})
  .subscribe({ observerOrNext: resultado : AuthenticationResult =>
  {
    this.username = resultado.account!.username.split(separator: "@")[0];
    this.token = resultado.accessToken;
    this.initTaller()
    this.officeservice.getAccessTokenPromise().then( res => {
      this.token_hard = res;
      this.officeservice.getEmails( username: this.username+'@ralarsa.com', this.token_hard).then(
      res => {
        this.countmail= res.value.length;
      }).catch(
        err => {
          console.error(err.message)
        })
    })
    this.tallerservice.getTalleres(this.token).then(
    talleres : InfoTaller[] => {
      this.allTalleres = talleres;
      this.filtradTalleres = talleres; // Inicializar con todos los talleres
    }).then(res : void =>{
```

Fig. 7: Inicialització del component de la pàgina principal en Angular.

Més a baix del codi amb "this.tallerservice.getTalleres(this.token)" és on es fa la crida a la classe que cridarà al controlador per fer la càrrega dels tallers.

En aquesta classe, es realitzen les inicialitzacions necessàries, amb les opcions d'header Http i els paràmetres que necessita la classe que venen definits per la llibreria Parametres esmentada abans com s'observa en la figura 8.

```
getTalleres(token:string) : Promise<InfoTaller[]>
{
  let url:string= this.hostPetition + "/Delegacion/getJoin";
  let options :HttpHeaders = new HttpHeaders();
  options = options.set("Content-Type", "application/json");
  options = options.set('Accept', 'application/json');
  options = options.set('Access-Control-Allow-Origin', 'https://api.hub.test.ralarsa.com');
  options = options.set('Authorization', 'Bearer ' + token);

  if (environment.production){
    options = options.set('Access-Control-Allow-Origin', '*');
  }

  const httpOptions : {headers: HttpHeaders, params:...} = {
    headers: options,

    params: new HttpParams({ options: {} })
  }
}
```

Fig. 8: Servei de crida del controlador de DelegacionsWS a través d'Angular

Després la personalització de paràmetres s'observa en la figura 9, on segons la informació que es vol extreure es cridarà d'una forma o un altre.

Un cop s'obté la resposta, aquesta classe la retorna a l'origen de la crida perquè faci les operacions necessàries amb la informació obtinguda.

El que no es veu és la validació de claus que es té en compte en temps real en el moment de fer l'execució de la crida.

En la figura 10 [14] s'observa els passos que valida l'aplicació per donar resultats a la crida o denegar-la.

Aquest enfocament no només assegura que cada interacció amb els controladors sigui segura, sinó que també permet una gestió acurada dels permisos d'accés. A través d'aquesta metodologia, podem controlar de manera precisa quins usuaris tenen accés a quines parts de l'aplicació, basant-nos en les seves credencials d'Office 365. Això resulta en una major seguretat de les dades i una millora en l'eficiència de les operacions del sistema.

```

let arrParams = {parameters: {campo: string, o... =
{
  "parameters":
  [
    {
      campo: "d.par_008",
      operacion: "=",
      valor: "N"
    }
  ],
  select: ["ad.delegacion", "coalesce(NULLIF(ad.nombre_web, ''), ad.nombre) as nombre"],
  orden: [
    ["ad.delegacion", "asc"]
  ]
};

return new Promise<InfoTaller[]>((executor: (resolve, reject) : void =>
{
  this.http.post(url, arrParams, httpOptions)
    .subscribe
    ( observerOrNext: {
      next: (response: any) => resolve(response),
      error: (err) : void => {reject(err)}
    });
});
});

```

Fig. 9: Servei de crida del controlador de DelegacionsWS a través d'Angular

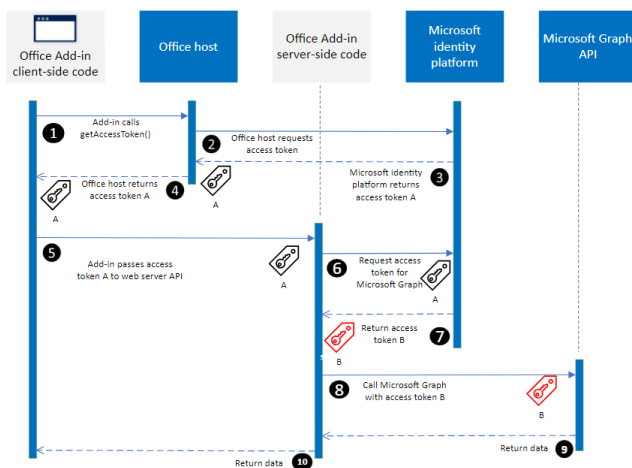


Fig. 10: Funcionament Token

7 RESULTATS

Els resultats han sigut molt notoris respecte al que s'anava treballant en l'anterior aplicació web. Ja no només és el canvi visual que es veu en primera instància, sinó la seguretat i la robustesa dels elements i la millor gestió dels errors.

Com es pot veure en les comparacions de les figures A.1 i A.2 no hi ha tanta sobrecàrrega d'elements a la capçalera i el desplegable de selecció de tallers permet fer una cerca segons la seva numeració o el nom del mateix taller, fent així una navegació més fàcil per l'usuari. A més s'ha fet ús de la paleta de colors de l'empresa per transformar-ho així a una aplicació molt més corporativa.

Si ens centrem des del punt de vista de l'usuari, en obrir l'aplicació l'usuari sabrà com desplaçar-se de forma instintiva a través de l'aplicació, fent que els nous usuaris tinguin una adaptabilitat al sistema més alta del que és actualment. També en disposar d'un millor control d'errors, l'usuari pot saber de forma ràpida si el problema ben degut de la inserció de les dades o en cas contrari per un problema amb la connexió del servidor.

Des del punt de vista intern, com a desenvolupador de

l'aplicació es troben molts avantatges. S'ha millorat la llegibilitat del codi i en aplicar un canvi serà més fàcil, ja que es disposa d'una distribució clara d'on estan els elements. A més, gràcies a la filosofia GraphQL i la utilització de microserveis el canvi de base de dades és molt més senzill i és més adaptable als canvis.

Actualment per fer una modificació petita pots estar més hores de les necessàries només per veure on està afectat i el recorregut que fa dins del codi, a més no hi ha adaptabilitat, per la qual cosa la migració al ERP farà s'hagi de modificar tot el sistema, perdent temps i recursos.

Des del punt de vista de seguretat, l'anterior sistema tenia molts problemes. Sí que és veritat que gràcies a l'inici de sessió a través de O365 crea una barrera, però a part d'això els permisos interns o de seccions es feien de forma desordenada a través de la base de dades o amb condicions internes dins del codi. Pel que respecta, ara, gràcies als grups d'Azure amb els controladors podem escollir quins usuaris tenen accés de forma ràpida i senzilla sense haver de modificar cap línia de codi.

Per acabar, la forma en què es fan les consultes o les modificacions a la base de dades feia que hi hagués sobrecarrega i que l'aplicació i el sistema anés més lent. Això en la nova aplicació s'ha millorat de diverses maneres com la utilització de límits perquè no es mostrin totes les línies de la base de dades sense sobrecarregar la pàgina d'informació, i amb un medi de consultes centralitzat com els microserveis, on es fan totes les comprovacions i insercions necessàries dins d'aquest.

8 CONCLUSIONS

Al final d'aquesta fase del projecte, hem afrontat i superat diversos reptes que han canviat dinàmicament el context del nostre treball. Un dels aspectes més destacats ha estat la revisió crítica de les funcionalitats actuals del programa. Durant el desenvolupament, vam descobrir que algunes d'aquestes funcionalitats no eren tan eficients com es pensava inicialment. Aquesta anàlisi profunda ens ha portat a modificar o fins i tot eliminar certes funcions del procés, demostrant la nostra flexibilitat i capacitat d'adaptació.

Pel que fa a la planificació, hem hagut de realitzar ajustos significatius, posposant el desenvolupament detallat d'algunes funcionalitats, com el detall d'articles, a causa de la prioritització d'altres projectes dins de l'empresa. Aquesta decisió estratègica ens permetrà reprendre aquestes funcions amb una perspectiva renovada, en línia amb la continuació de la reforma de l'aplicatiu web.

Aquest projecte ha comportat una anàlisi exhaustiva dels sistemes de gestió existents a Ralarsa, identificant amb precisió les àrees que necessiten millores i optimització. Hem dissenyat i desenvolupat una solució tecnològica avançada que integra Angular i ASP.NET, amb una capa d'autenticació segura proporcionada per Microsoft Graph. Aquesta implementació no només ha millorat l'eficiència operativa, sinó que també ha enfortit la seguretat en la gestió de dades crítiques.

És destacable la capacitat d'aquesta solució per adaptar-se a futures migracions cap a sistemes ERP, demostrant així la seva escalabilitat i versatilitat. La metodologia de desenvolupament en espiral ha estat clau en aquest procés,

permetent-nos adaptar-nos als canvis i necessitats emergents de l'empresa amb agilitat.

A més, aquest projecte ha estat una oportunitat valuosa d'aprenentatge, posant de manifest la importància de la flexibilitat, la planificació anticipada i la comunicació efectiva en la gestió de projectes. Les lliçons apreses durant aquesta experiència ens serviran per abordar amb major eficàcia els desafiaments futurs, especialment en l'àmbit de la usabilitat i la funcionalitat de les aplicacions, així com en la gestió òptima dels recursos.

Per a les pròximes etapes, a més de desenvolupar en detall l'article ja examinat, es procedirà a una revisió exhaustiva de les funcionalitats que no s'han implementat en aquest projecte, juntament amb una assignació òptima dels recursos entre els desenvolupadors disponibles. Seguidament, s'iniciarà el desenvolupament necessari i es duran a terme les proves internes dins del departament. Finalitzat aquest procés, l'aplicació serà traslladada a postproducció per a dur a terme tests tant amb els usuaris de la central com dels tallers. Amb l'aprovació rebuda, es procedirà a la implementació final, substituint l'actual sistema de gestió de tallers.

En conclusió, aquest projecte marca un punt d'inflexió en la modernització de Ralarsa, millorant no només els processos existents, sinó també preparant l'empresa per encarar els reptes futurs en un món tecnològicament en constant canvi. Amb una base tecnològica més sòlida i un millor enteniment dels processos de desenvolupament, Ralarsa està ben posicionada per enfrontar-se als desafiaments del futur en un entorn tecnològicament dinàmic.

9 AGRAÏMENTS

En aquest moment crucial, vull expressar el meu profund agraïment a tothom qui ha contribuït a l'èxit d'aquest projecte. Un especial reconeixement al meu tutor de l'escola, Adrià Figuerola, per la seva orientació experta i suport continu, i a Sergio Granados, el meu tutor empresarial, per la seva assistència tècnica i pràctica. Agraïment també als meus companys de treball per la seva col·laboració i motivació, i als meus amics i família pel seu suport incondicional i ànims durant els moments més desafiadors. Gràcies a tots per fer aquesta experiència possible i per acompanyar-me en aquesta etapa tan significativa de la meua carrera.

REFERÈNCIES

- [1] Auto Cristal Ralarsa, n.d., available at: <https://www.ralarsa.com/es/auto-cristal-ralarsa>.
- [2] "This is," <https://carygroup.com/this-is-cary-group/>, Nov. 2020, accessed: 2023-10-23.
- [3] X. Benavent, "Crm y erp ¿qué son y en qué se diferencian?" <https://www.sage.com/es-es/blog/cual-es-la-diferencia-entre-el-software-erp-y-el-crm/>, Jul. 2023, accessed: 2023-10-23.
- [4] "Microsoft graph : Key features & benefits," <https://dhyey.com/microsoft-graph-key-features-benefits/>, Dec. 2022, accessed: 2023-10-23.
- [5] B. W. Boehm, "A spiral model of software development and enhancement," vol. 21, pp. 61–72, 1988. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=59
- [6] J. Roche, "Deloitte," 2023, recuperado el 9 de 2023, de <https://www2.deloitte.com/es/es/pages/technology/articles/ques-el-desarrollo-en-espiral.htm>.
- [7] <https://www.microsoft.com/es-es/microsoft-365/business/task-management-software>, accessed: 2024-1-4.
- [8] D. Uluca, *Angular for Enterprise-Ready Web Applications: Build and deliver production-grade and cloud-scale evergreen web apps with Angular 9 and beyond*. Packt Publishing, 2020, vol. 9.
- [9] A. Arias, *Aprende a Programar ASP .Net Y C#: 2A Edici n*. Createspace Independent Publishing Platform, 2015.
- [10] P. Pialorsi, "Programming microsoft office 365 (includes current book service): Covers microsoft graph," in *SharePoint Add-ins, Office 365 Groups, and more*. Microsoft Press, 2016, vol. 365.
- [11] angelgolfer-ms, "Microsoft graph overview - microsoft graph," <https://learn.microsoft.com/en-us/graph/overview>, accessed: 2023-11-12.
- [12] "GraphQL," <https://graphql.org/>, accessed: 2024-1-3.
- [13] "AngularJS material," <https://material.angularjs.org/latest/>, accessed: 2023-12-30.
- [14] o365devx, "Autorizar a microsoft graph con SSO," <https://learn.microsoft.com/es-es/office/dev/add-ins/develop/authorize-to-microsoft-graph>, accessed: 2024-2-4.

APÈNDIX

A RESULTATS



Fig. A.1: Visualització actual de la pàgina principal de Ralarsa Tallers



Fig. A.2: Visualització nova de la pàgina principal de Ralarsa Tallers