

MEMORIA DEL PROYECTO – ERP 2022 - 2023

24 OCTUBRE - 13 DICIEMBRE

ERP

Tutor:

José Luis Meriño

Creado por:

Katherine Almache Riofrio & Jorge Casillas

Contenido

1. DEFINICIÓN DEL SISTEMA	3
1.1 ANÁLISIS DE LA UTILIDAD/NECESIDAD DEL SISTEMA. PROYECCIÓN A NIVEL EMPRESARIAL.	3
1.2 IDENTIFICACIÓN DEL ALCANCE DEL SISTEMA.	4
1.3 IDENTIFICACIÓN DEL ENTORNO TECNOLÓGICO Y DE LAS TECNOLOGÍAS A UTILIZAR.	7
1.4 PLANIFICACIÓN DE LAS FASES DEL DESARROLLO DEL PROYECTO.	8
I. MÉTODO UTILIZADO PARA EL DESARROLLO DEL SOFTWARE SEÑALANDO LAS FASES DE DESARROLLO DEL PROYECTO (ANÁLISIS, DISEÑO, IMPLEMENTACIÓN, PRUEBAS...) Y SU ORGANIZACIÓN TEMPORAL.	8
II. CICLO DE VIDA UTILIZADO. EN EL CASO DE QUE SE UTILICE ALGUNA METODOLOGÍA CONCRETA (SCRUM, MÉTRICA...) TAMBIÉN SE ESPECIFICARÁ EN EL DOCUMENTO.	11
2. ANÁLISIS DEL SISTEMA	12
2.1 ESPECIFICACIÓN DE REQUISITOS.	12
2.2 ESPECIFICACIÓN DE ESTÁNDARES Y NORMAS DE APLICACIÓN EN EL PROYECTO.	12
3. DISEÑO DEL SISTEMA.	15
3.1 PLANIFICACIÓN DE LAS FASES DEL DESARROLLO DEL PROYECTO.	15
3.2 DISEÑO DEL MODELO DE DATOS.	16
3.3 OTROS DIAGRAMAS (OPCIONAL - SI HAY ALGÚN ASPECTO QUE CONVIENE DETALLAR EN DIAGRAMAS DE CASOS DE USO, CLASES, ETC).	18
4. ASPECTOS RELEVANTES DE LA IMPLEMENTACIÓN.	20
5. PRUEBAS DE ACCESIBILIDAD Y USABILIDAD.	20
6. DESPLIEGUE DEL PROYECTO.	25
7. PROBLEMAS ENCONTRADOS.	27
8. CONCLUSIONES. (EVALUACIÓN CRÍTICA DE LOS RESULTADOS OBTENIDOS. PROPUESTAS DE MEJORAS Y DESARROLLOS FUTUROS).	28
9. BIBLIOGRAFÍA.	28

1. Definición del sistema

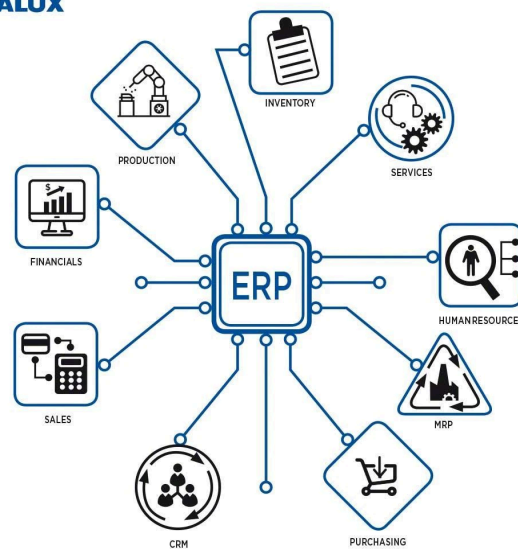
1.1 Análisis de la utilidad/necesidad del sistema. Proyección a nivel empresarial.

Nuestro proyecto a presentar sería la creación de nuestra propia ERP. Por lo tanto, desarrollamos nuestro un sistema de planificación de recursos empresariales. Él cual cuenta con lo necesario para ser una aplicación web, capaz de generar encuestas, almacenar datos, organizarlos de manera óptima y al final mostrar a partir de una gráfica.

Además está creada sobre la versión frameworkd ASP.NET MVC, dando forma a páginas web de carga más rápida y flexibles a la hora de navegar en ellas. Siendo esta herramienta de las más utilizadas actualmente en mundo empresarial, cerca del 25% las páginas creadas por empresas utilizan ASP.NET.

Asimismo, nuestro producto se aprovecha del auge del cloud computing, lo cual significa utilizar servicios informáticos como almacenamiento, software, redes y bases de datos a través de Internet. Utilizando un servidor externo conectado a nuestro modelo de aplicación .NET.

Esta nueva forma de acceder y almacenar recursos, es muy útil para mejorar la productividad de las empresas. En primer lugar no necesita de espacio de almacenamiento físico, además tiene la posibilidad de acceder a los recursos necesarios desde cualquier lugar, mejorando el tiempo de respuesta a la hora de tomar decisiones y la flexibilidad de acción frente a imprevistos. Todo ello, mencionado anteriormente reduce enormemente el coste de operaciones, mejorando positivamente el balance económico de la empresa.

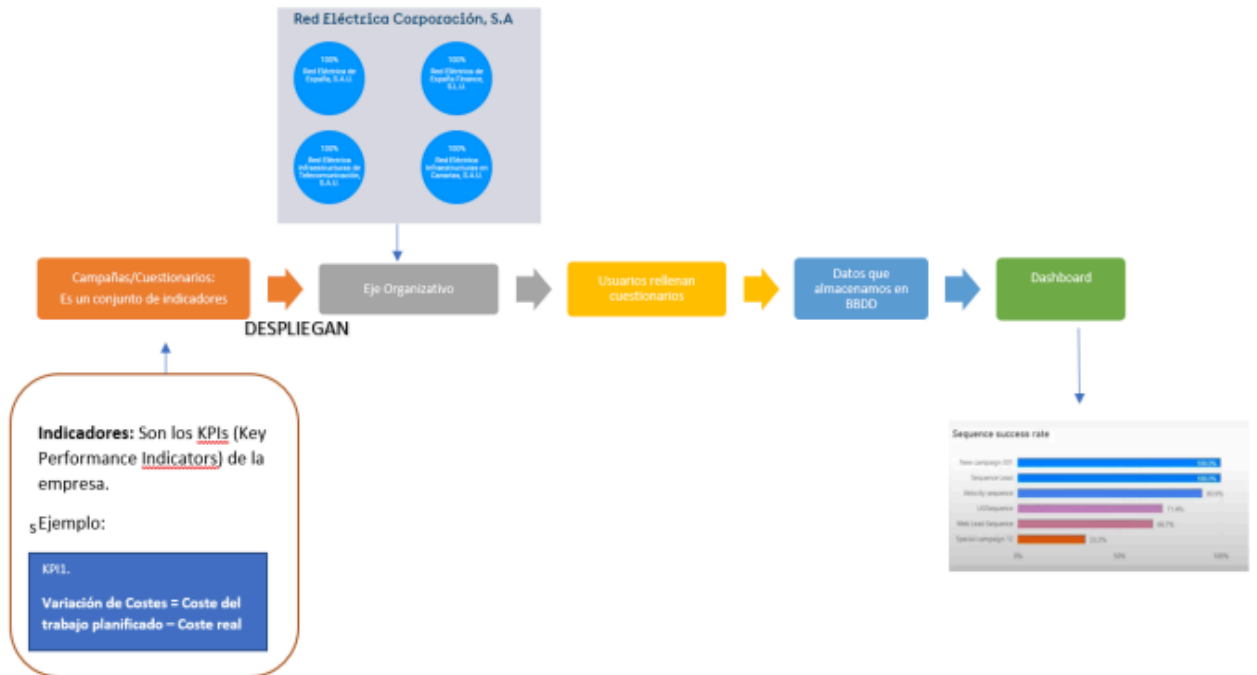


1.2 Identificación del alcance del sistema. (Objetivos, requisitos iniciales, algún diagrama inicial que represente el sistema...).

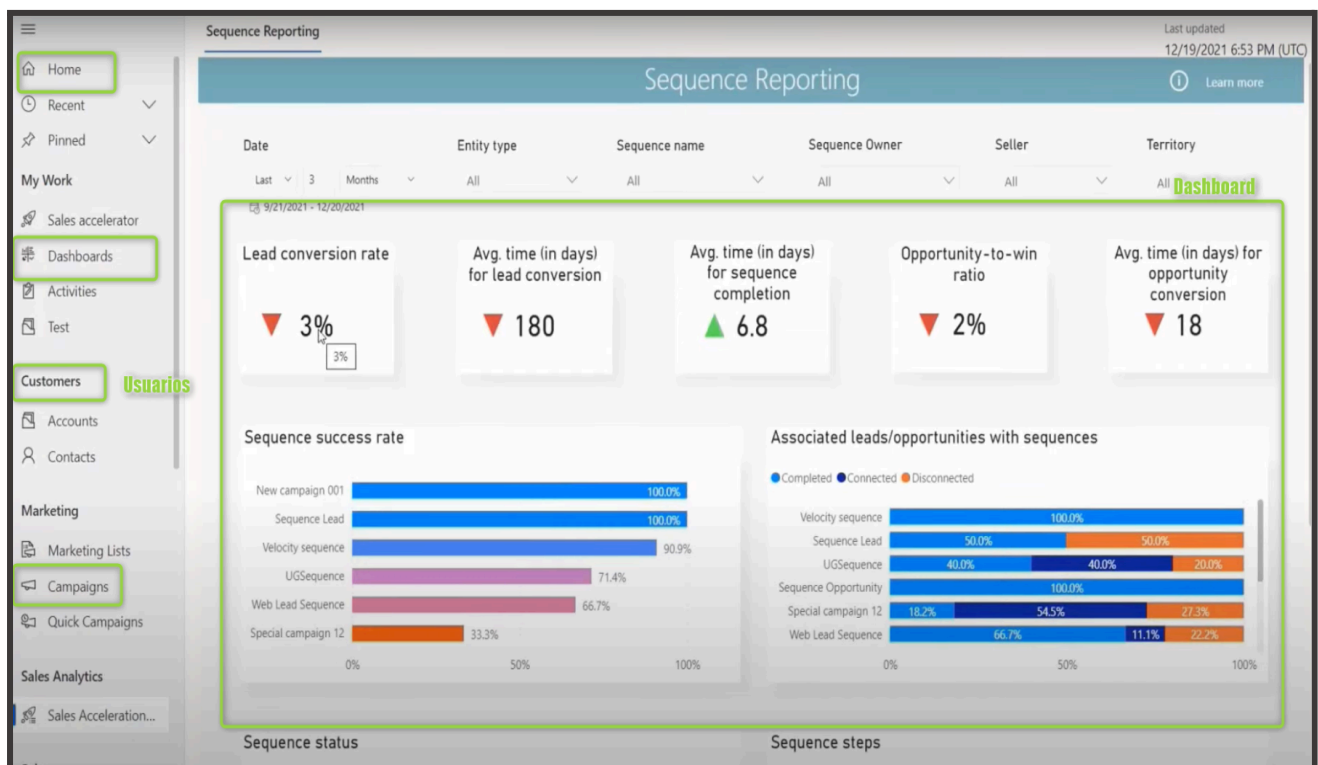
Nuestro objetivo como hemos mencionado antes se trataría de la creación de nuestra propia ERP, es decir, nuestro propio sistema de planificación de recursos. Durante la fase de planificación marcamos como requisito inicial, que el proyecto debería de poder mostrar datos registrados anteriormente por medio de encuestas y campañas, además de poder modificarlos y añadir nuevos. Durante la fase de desarrollo nos enfocamos en hacer un diagrama que representa el modelo de datos que nos guiaría como plantilla a la hora de desarrollar el proyecto. Creando la aplicación con ASP.NET, y dividiendo el proyecto en base al Modelo-Vista-Controlador. Por todo esto nuestra página se dividirá en dos áreas lógicas: el back-end; encargado de procesar la información para que funcione la web y el front-end; encargado de mostrar la interfaz web al usuario.

Además para poder almacenar datos necesitábamos una base de datos, nos decantamos por SQL server ya que era conocido para nosotros. Creando y usando tablas de datos y procedimientos almacenados.

Módulo Métricas



Viendo la imagen superior, encontramos lo que sería la línea a seguir. Crear campañas y cuestionarios alojados en eje organizativo y que serían rellenados por los usuarios con acceso y los datos serían almacenados en bbdd para su manejo posterior por parte del usuario. Por último, una gráfica mostraría los datos introducidos con anterioridad.



(Ejemplo de una ERP terminada)

1.3 Identificación del entorno tecnológico y de las tecnologías a utilizar.

Nuestra aplicación web fue creada en Visual Studio en base a ASP.NET, siguiendo el modelo arquitectónico de MVC, es decir de modelo vista controlador. Por lo tanto, cuenta con las tres partes específicas del MVC, que son las siguientes: controlador, vista y modelo. En consonancia con el MVC, nuestra aplicación contará con una parte front-end y back-end.

En primer lugar resaltar un lenguaje que forma parte del frontend pero que se ejecuta a la vez, dentro del servidor es cshtml. Se trata de un archivo html c#, que el motor Razor utiliza para representar archivos de la página web en el navegador siendo similar al ASP.NET pero enfocado a tareas más complejas como acceder a bases de datos y renderizar vistas complejas. Este lenguaje formaría parte de la vista del MVC y será la base para crear nuestra aplicación.

Además de este otros lenguajes usados en front-end son: javascript y css. Al usar javascript durante el desarrollo de la web utilizamos JSON y JQuery, perfectos para conectar html y javascript, en base a funciones como Ajax por ejemplo.

Por otra parte en back-end se encuentran la parte modelo y controlador, donde se utiliza el lenguaje c# propio de Visual Studio, creado por Microsoft como parte de la plataforma .NET, en forma de API.

Por último mencionar el uso de SQL como lenguaje diseñado para gestionar base de datos. Al usar un servidor externo, usamos este lenguaje para gestionar nuestra información almacenada allí.

1.4 Planificación de las fases del desarrollo del proyecto.

- i. Método utilizado para el desarrollo del software señalando las fases de desarrollo del proyecto (Análisis, Diseño, Implementación, Pruebas...) y su organización temporal:
 - **FASE DE PLANIFICACIÓN:** En esta fase inicial del proyecto determinamos el ámbito del proyecto en nuestro caso la creación de una ERP. Realizamos un estudio de viabilidad en base al mercado actual de ERP junto a un análisis de los riesgos asociados al proyecto, y una estimación del coste de tiempo del proyecto. Por último el desarrollo de lo que sería una planificación temporal.
 - **FASE DE ANÁLISIS:**
 1. Detectamos la necesidad de una base de datos relacionada, por ello vamos a desarrollar un Diagrama Entidad Relación (E/R) que nos permitirá visualizar la relación entre los datos para poder almacenar los datos que necesitamos. Así como sus cardinalidades.
 2. Los elementos que forman nuestro diagrama E/R:
Detectamos la necesidad de una base de datos relacionada, por ello vamos a desarrollar un Diagrama Entidad Relación (E/R) que nos permitirá visualizar la relación entre los datos para poder almacenar los datos que necesitamos. Así como sus cardinalidades.
 3. Los elementos que forman nuestro diagrama E/R:
 - Entidad
 - Relación: definimos las cardinalidades y estas pueden ser:
 - 1:1 □ Se nos pueden presentar 3 casos a la hora de hacer la transformación a tablas:
 - o Si ambas entidades tienen cardinalidad (1,1), en cuyo caso fusionamos ambas entidades generando una única tabla cuya clave primaria puede ser cualquiera de las dos claves primarias que la forman.
 - o Si una de las entidades tiene cardinalidad (0,1) y la otra (1,1), propagamos la PK de la entidad (1,1) pasando a ser esta foránea en la de (0,1).
 - o Si ambas son (0,1) se transforma en una tabla nueva cuya clave primaria será la concatenación de las claves primarias que también serán foráneas.

- 1:N □ Un registro de una entidad (X,1) se relaciona con varios registros de la entidad de cardinalidad (X,N). Para formarla nos tenemos que fijar en la entidad de cardinalidad (x,1) en su valor X que puede ofrecer dos casos:
 - $X=1$ □ La relación no se transforma en una tabla. Se propaga la clave primaria de la entidad de cardinalidad (1,1) como foránea a la entidad de cardinalidad (X,N). Propagar implica que la clave primaria de la entidad (1,1) pasa a ser foránea en la entidad de cardinalidad (x,N)
 - $X=0$ □ Necesitamos generar una tercera tabla. Tendrá una clave primaria que a la vez será foránea de la entidad que tiene de cardinalidad máxima N. Los atributos de la relación si existen pasan a ser atributos de la tercera tabla construida. La clave primaria de la entidad de cardinalidad 0,1 pasa a la nueva tabla como foránea.
- N:M □ da lugar a una nueva tabla tendrá como atributos los que tuviera la relación y como PK la concatenación de las PK de las entidades relacionadas y además la PK también será foránea cada uno de ellos, en la nueva tabla
- Atributo: son propiedades de una entidad
 - Atributo principal/PK
 - Atributo Foránea/FK

Será una base de datos que gestione el despliegue de encuestas en distintas partes de la organización. Por ello detectamos la siguiente información:

1. Identificamos los datos o tablas. → listas de tablas**
2. Identificamos los elementos que conforman estas tablas - serían los atributos

- FASE DE DISEÑO:

1. En la fase de diseño se han de estudiar posibles alternativas de implementación para el sistema de información que hemos de

construir y se ha de decidir la estructura general que tendrá el sistema (su diseño arquitectónico). En nuestro caso nos decantamos por Visual studio y el uso de ASP.NET para el desarrollo de nuestra web.

2. Además es necesario abordar el diseño de la base de datos, y sus elementos. Utilizamos como referencia las entidades sacadas del diagrama de relación.
 3. También empezamos diseñando cómo será la interfaz de usuario del sistema y los distintos componentes en que se descomponen las aplicaciones. Nos fijamos en proyectos de ERP completados y como lucen al final.
- **FASE DE IMPLEMENTACIÓN:**
 1. Creamos la base de datos en SQL.
 2. Creamos las distintas partes que tendrá nuestra aplicación web, desde distintas capas y la creación de objetos para su futura implementación.
 - **FASE DE PRUEBA:**
 1. Después de terminar aparentemente, nos dedicamos a probar la aplicación y corregir fallos o realizar mejoras para la correcta funcionalidad de la app.

Fases	Nº	Descripción	Comentarios
Análisis		Base de datos relacional	
		Diagrama E/R: Atributos Entidades 1 Relaciones	01 - ASP.NET - Conectar una aplicación web con sql - VB - YouTube
	2	Script para base de datos	
		Formularios	Roles/Grupos:
	1	Formulario de Acceso Sistema con usuario y contraseña y visibilidad. Dependiendo de que Rol y alcance (Entidad)	Contribuidor: que será aquel que tiene que llenar las campañas/encuestas puede ver, editar, no borrar y no podrá editar una vez enviado el formulario. Supervisor: ver, modificar y aprobar la encuesta. Administrador: ver, crear, editar, borrar.
	2	Formulario de datos al inicio: muestra datos en reports	
	3	Formulario eje organizativo -->	
	4	Formulario de Geografía	Formulario tabla de ajustes con niveles
		Formularios específicos del módulo de Métricas:	
	5	Formulario de Indicadores: recoge el tipo de datos a recoger	Ajustes formulario: Tipo de dato --> respuesta yes, no, número, indicador calculado, de entrada manual. Unidades: unidades en se reporta
	6	Formulario de Campañas: es en el que indique las entidades dónde se va a desplegar según perfiles.	
		Formularios Usuarios:	
	7	Formulario usuarios: muestra todos los usuarios	Ajustes: grupos/roles

(Tabla Fase de análisis)



-
- ii. **Ciclo de vida utilizado.** En el caso de que se utilice alguna metodología concreta (SCRUM, MÉTRICA...) también se especificará en el documento.

Nosotros debido a nuestro reducido tamaño nos decantamos por la metodología Agile que se usa en el desarrollo de software. Esta se centra en la implementación rápida de un equipo eficiente y flexible para planear el flujo de trabajo.

Agile brinda la capacidad de elegir la mejor opción en cada situación sin comprometer el proyecto, potenciando la autonomía y flexibilidad de los implicados en el proyecto.



2. Análisis del sistema

2.1 Especificación de requisitos.

Los requisitos necesarios para el correcto funcionamiento de la aplicación son en primer lugar tener instalado Visual Studio 2019, dentro de la aplicación de instalación, seleccionar estas cargas de trabajo: desarrollo de ASP.NET y web, desarrollo de escritorio de .NET, desarrollo para el escritorio con C# almacenamiento y procesamiento de datos.

Debe tener también acceso a internet para poder acceder a la BBDD alojada en la nube. Además ha de estar registrado en nuestra base de datos para poder acceder a la web desde el Login.

2.2 Especificación de estándares y normas de aplicación en el proyecto.

1. Procesos Principales

a. Adquisición:

- i. El adquirente usará la aplicación erp para uso personal y en desarrollo de su actividad empresarial.
- ii. El adquirente se trataría del instituto IES El Lago.

b. Suministro:

- i. El proveedor daría un programa rpt al uso del consumidor.
- ii. El proveedor daría un programa rpt al uso del consumidor.

c. Desarrollo:

- i. El producto en cuestión sería un sistema de planificación de recursos empresariales.

d. Operación:

- i. El producto será utilizado en base a las condiciones que fue creado. Es decir, para hacerse cargo de distintas operaciones internas de una empresa.

e. Mantenimiento:

- i. No habrá mantenimiento futuro por parte nuestra.

2. Procesos de Soporte:

- a. Documentación:
 - i. El producto en cuestión sería un sistema de planificación de recursos empresariales. Por lo tanto se dedicará a operaciones internas de la empresa: desde producción a distribución o incluso recursos humanos.
- b. Administración de configuración:
 - i. Informes.
 - ii. Organizaciones.
 - iii. Usuarios.
 - iv. Encuestas.
- c. Aseguramiento de calidad:
 - i. Creado en base a Visual Studio, ASP.Net, por lo que cuenta con gran seguridad.
- d. Verificación:
 - i. No tenemos firma de verificación.
- e. Validación:
 - i. El adquiridor informará de que la aplicación no cumple con su cometido si llega a ser el caso.
- f. Revisión Conjunta:
 - i. Proceso de examinación de la aplicación.
- g. Auditoría:
 - i. Después de que el instituto delibere sobre nuestra aplicación tendremos correspondencia por mensaje.
- h. Resolución de problemas:
 - i. Proceso de verificación de procesos asociados al buen desarrollo del producto.

3. Procesos Organizados:

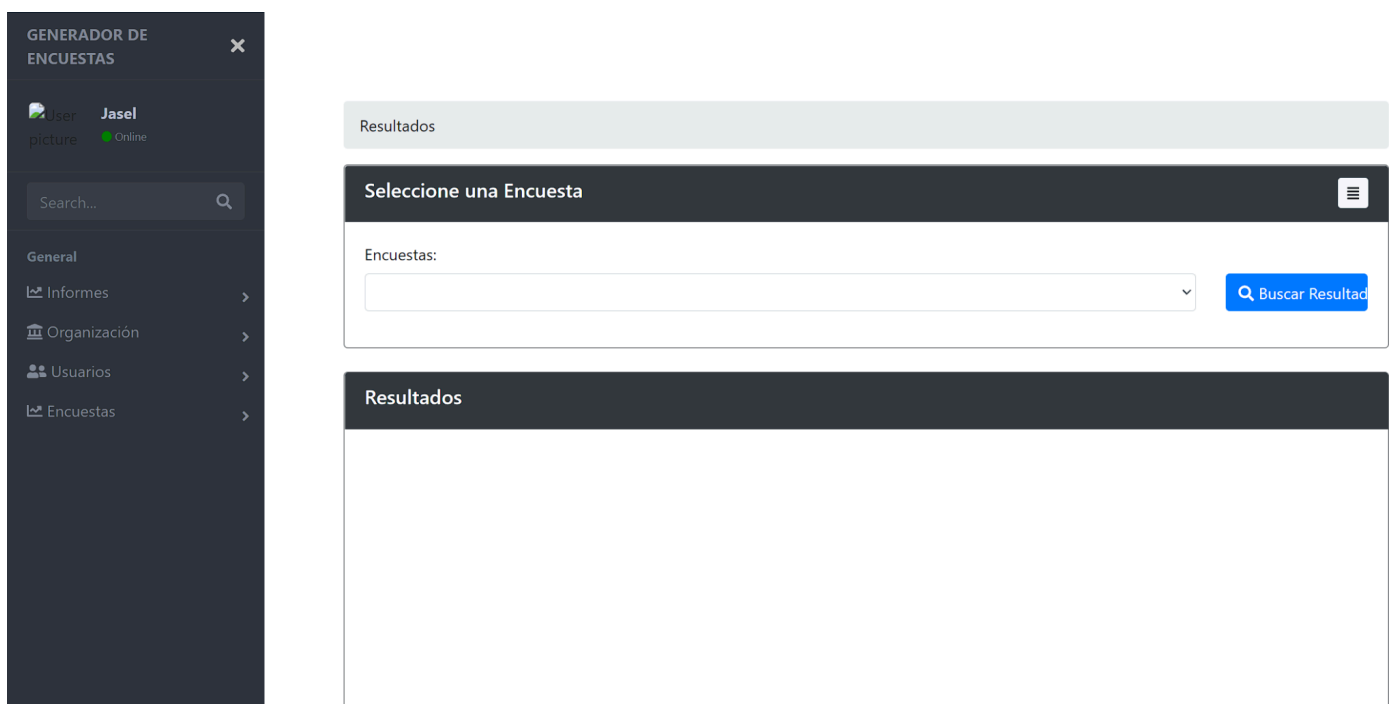
- a. Gestión:
 - i. Se dedicará a operaciones internas de la empresa: desde producción a distribución o incluso recursos humanos.
- b. Infraestructura:
 - i. Desarrollado en arquitectura MVC. Por lo tanto con área de back y front.
- c. Mejora:
 - i. Aportar gráficas de datos para acompañar los datos.
- d. Recursos Humanos:
 - i. Somos dos programadores.



3. Diseño del sistema.

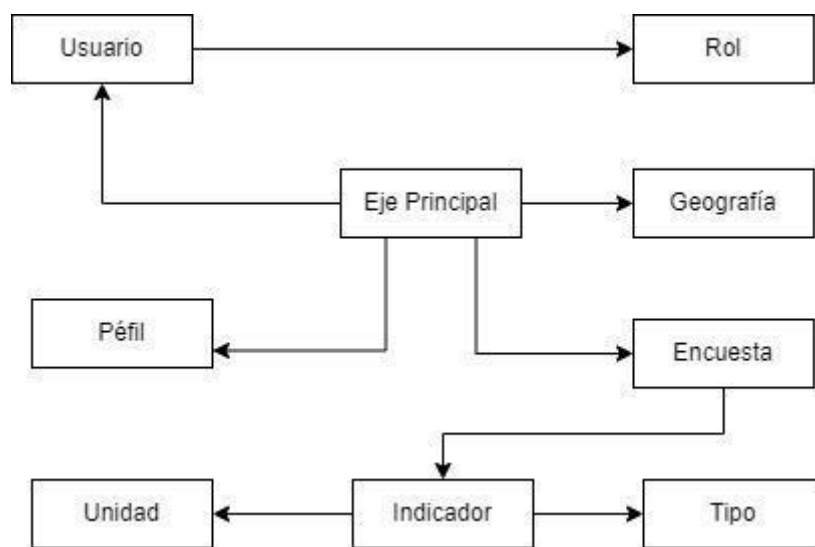
3.1 Diseño de interfaces de usuario.

Al empezar nos decantamos por tonalidades de color cálidas, para hacer nuestra interfaz más llamativa al usuario, pero al final nos alejamos de esta idea y utilizamos tonalidades más frías. Los tonos oscuros y grisáceos son idóneos para dar una imagen de elegancia y sobriedad. Colores demasiado fuertes podrían causar incomodidad y molestias al que observe la página durante un tiempo considerable.

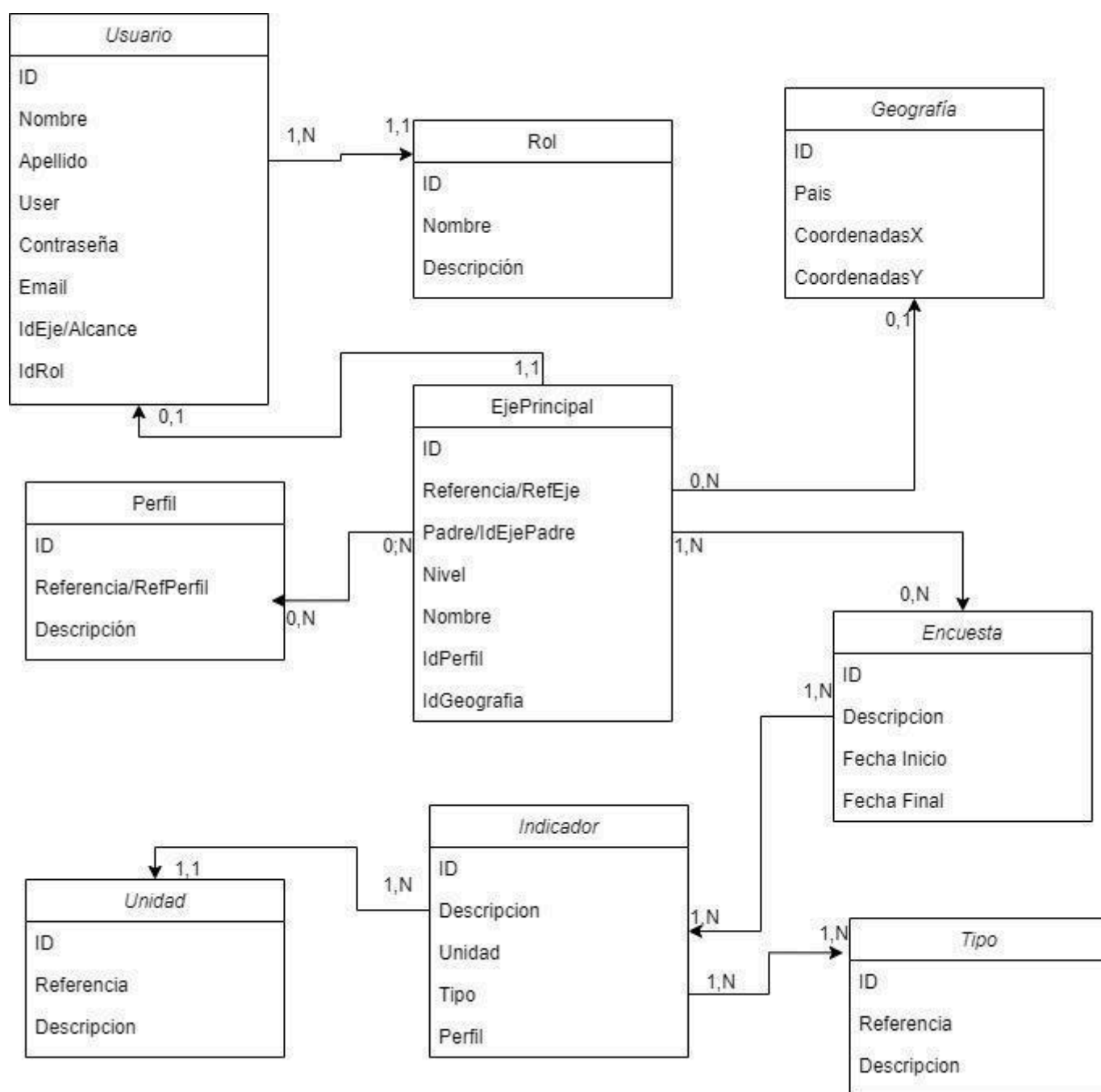


3.2 Diseño del modelo de datos.

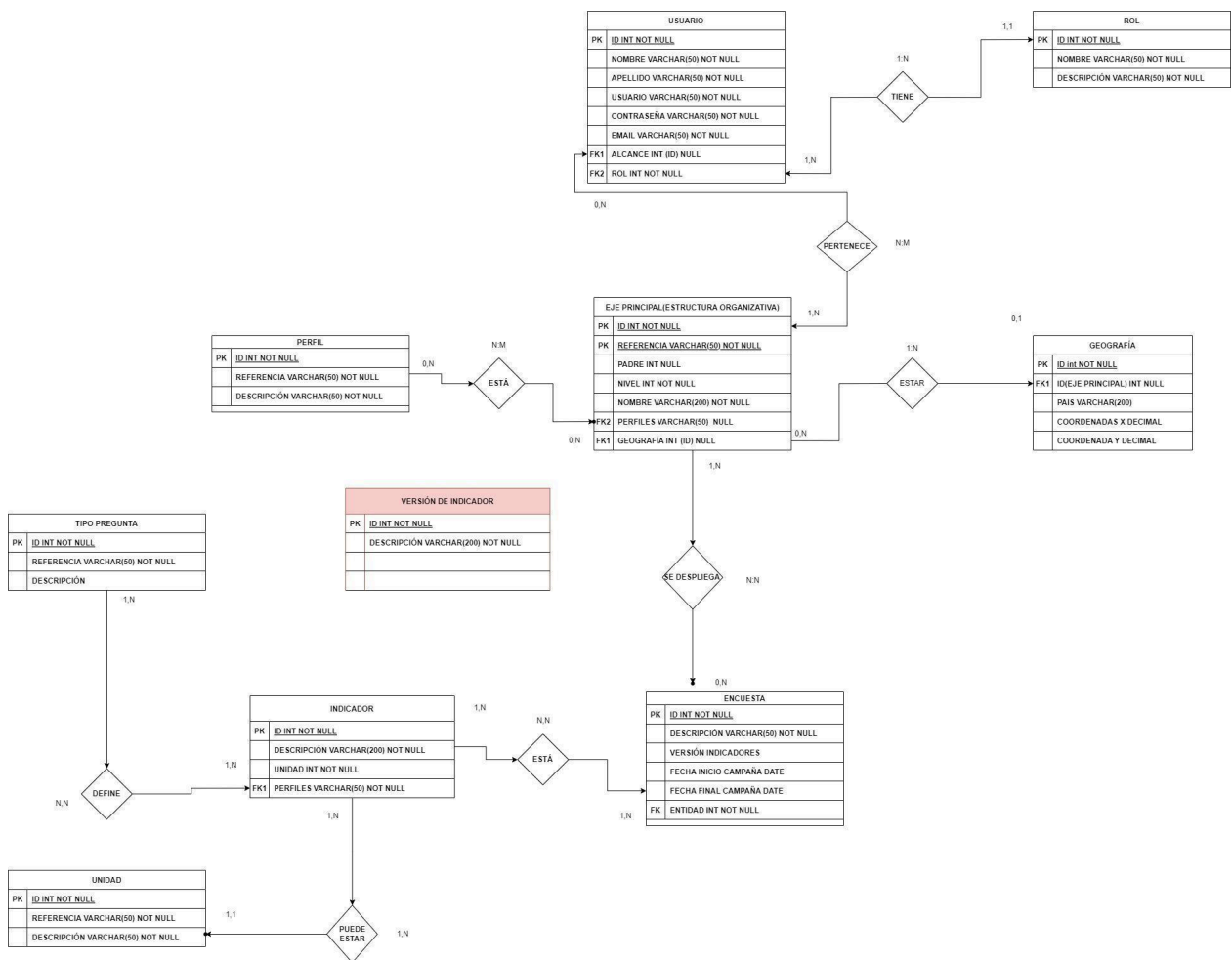
- Modelo de datos conceptual
- Modelo de datos lógico
- Modelo de datos físico



(Modelo conceptual)

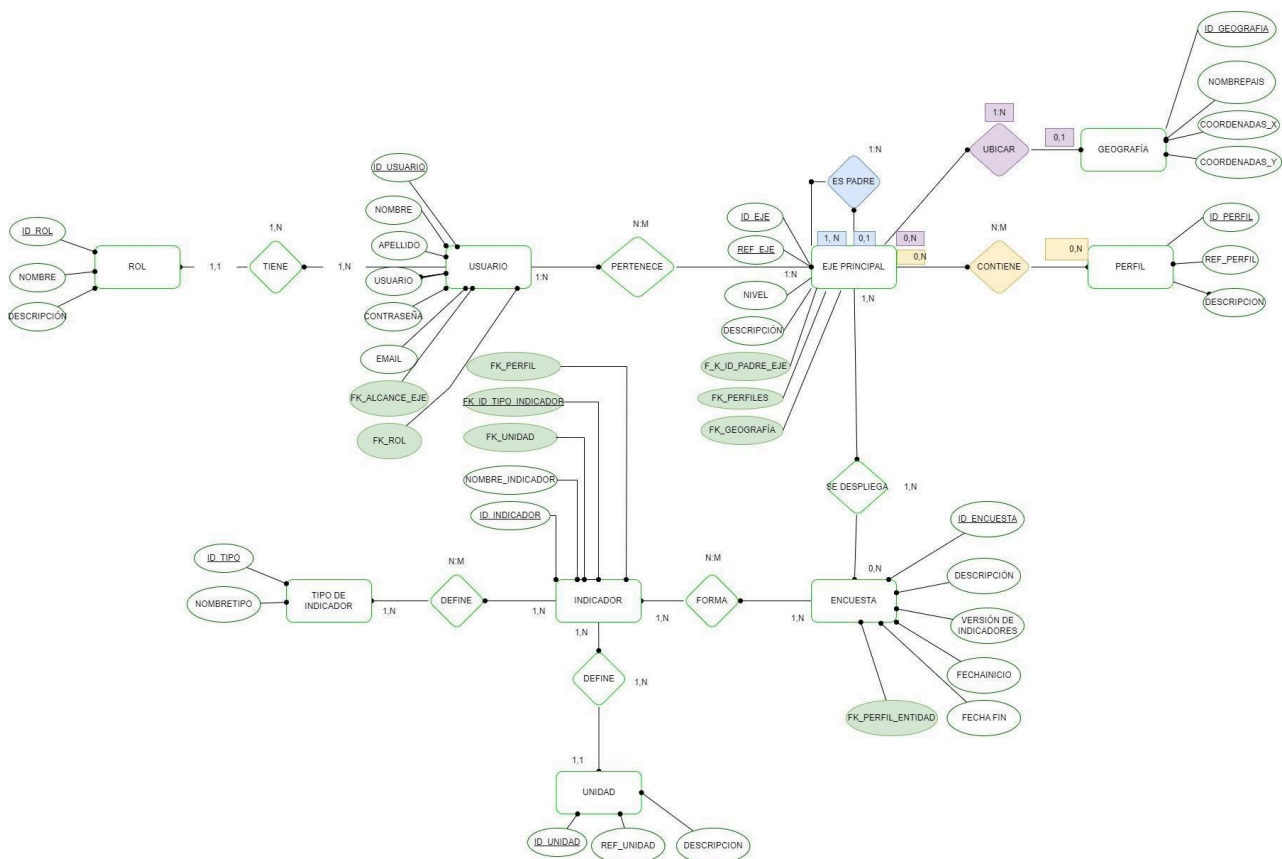


(Modelo de datos lógico)



(Modelo de datos físico)

3.3 Otros diagramas (Opcional - si hay algún aspecto que conviene detallar en diagramas de casos de uso, clases, etc).



(diagrama entidad-relación)

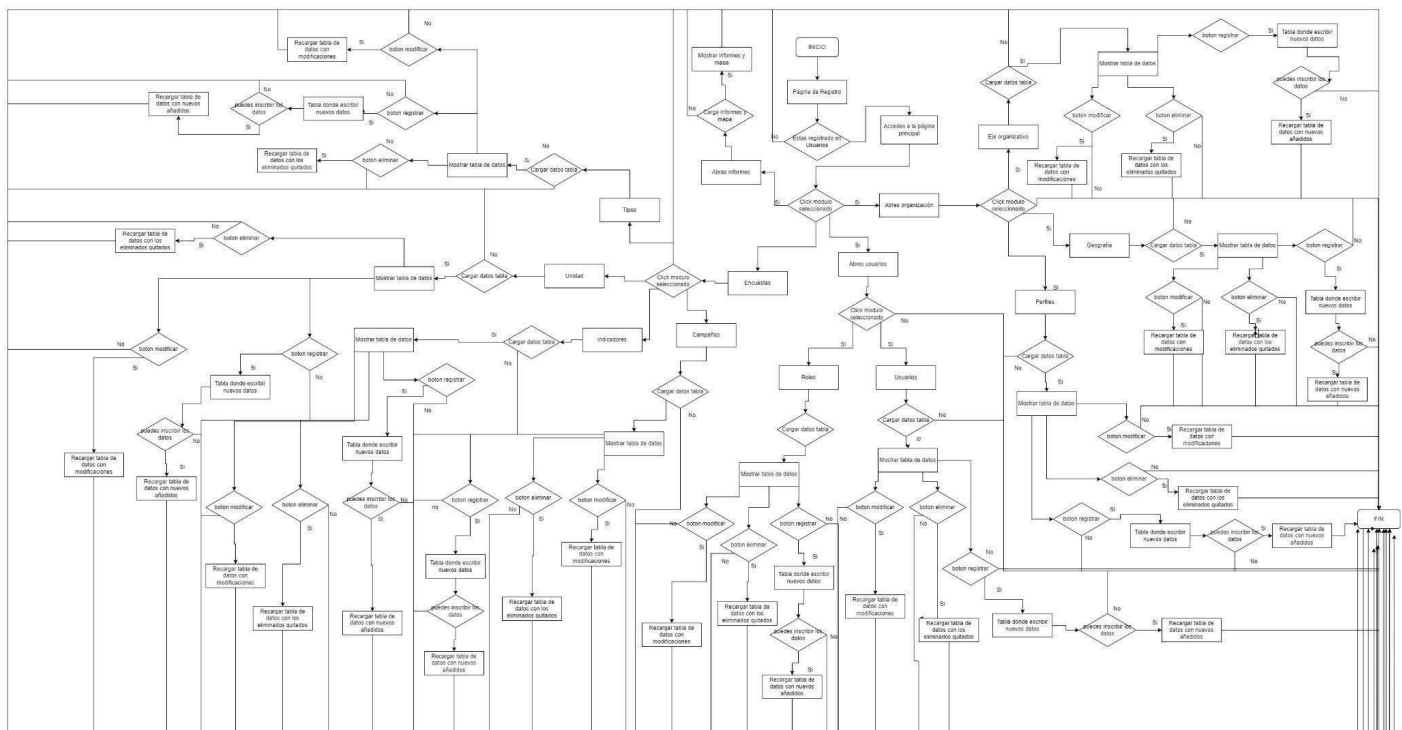
4. Aspectos relevantes de la implementación.

Como aspectos relevantes del proyecto, sería el uso de procedimientos para manejar de forma más sencilla la base de datos, junto con el uso de un servidor externo. Además resaltar que la aplicación está construida sobre la arquitectura MVC, dando lugar a distintas capas y formando comunicaciones entre front y back. Por último mencionar el uso que hemos dado de la plataforma online de alojamiento, Github. Nos facilitó trabajar de forma más individual, rápida y con la capacidad de trabajar en base a ramas creadas a partir del modelo principal a la que subíamos nuestros cambios sin afectar al trabajo del compañero.

5. Pruebas de Accesibilidad y Usabilidad.

Caja blanca:

Diagrama de flujo



Casos de prueba:

Número de caminos	Caso de Prueba	Resultado esperado
1	1	No está registrado
1,2	2	No funcionan los módulos
1,2,3	3	Accedes a organización pero no as sus subcategorías
1,2,3,4	4	Accedes a organización. Abres eje organizativo no carga tabla
1,2,3,4,5	5	Accedes a organización. Abres eje organizativo, carga tabla y la muestra
1,2,3,4,5,6	6	Accedes a organización. Abres eje organizativo, carga tabla y la muestra, además deja registrar
1,2,3,4,5,6,7	7	Accedes a organización. Abres eje organizativo, carga tabla y la muestra, además deja registrar y recarga la tabla
1,2,3,4,5,8	8	Accedes a organización. Abres organizativo, carga tabla y la muestra, además deja modificar y recarga la tabla
1,2,3,9	9	Accedes a organización. Abres geografía no carga tabla
1,2,3,10,11	10	Accedes a organización. Abres geografía, carga tabla y la muestra

1,2,3,10,11,12	11	Accedes a organización. Abres geografía y carga tabla y la muestra, además deja registrar
1,2,3,10,11,12,13	12	Accedes a organización. Abres eje organizativo, carga tabla y la muestra, además deja registrar y recarga la tabla
1,2,3,10,11,14	13	Accedes a organización. Abres geografía, carga tabla y la muestra, además deja eliminar y recarga la tabla
1,2,3,10,11,15	14	Accedes a organización. Abres geografía, carga tabla y la muestra, además deja modificar y recarga la tabla
1,2,3,16	15	Accedes a organización. Abres perfiles no carga tabla
1,2,3,16,17	16	Accedes a organización. Abres perfiles, carga tabla y la muestra
1,2,3,16,17,18	17	Accedes a organización. Abres perfiles , carga tabla y la muestra, además deja registrar
1,2,3,16,17,18,19	18	Accedes a organización. Abres perfiles, carga tabla y la muestra, además deja registrar y recarga la tabla
1,2,3,16,17,20	19	Accedes a organización. Abres perfiles, carga tabla y la muestra, además deja eliminar y recarga la tabla
1,2,3,16,17,21	20	Accedes a organización.

		Abres perfiles, carga tabla y la muestra, además deja modificar y recarga la tabla
1,2,22	21	Accedes a usuarios pero no as sus subcategorías
1,2,22,23	22	Accedes a usuarios.Abres usuarios no carga tabla
1,2,22,23,24	23	Accedes a usuarios. Abres usuarios, carga tabla y la muestra
1,2,22,23,24,25	24	Accedes a usuarios. Abres usuarios, carga tabla y la muestra, además deja registrar
1,2,22,23,24,25,26	25	Accedes usuarios. Abres usuarios, carga tabla y la muestra, además deja registrar y recarga la tabla
1,2,22,23,24,27	26	Accedes usuarios. Abres usuarios, carga tabla y la muestra, además deja eliminar y recarga la tabla
1,2,22,23,24,28	27	Accedes a usuarios. Abres usuarios, carga tabla y la muestra, además deja eliminar y recarga la tabla
1,2,22,29	28	Accedes a usuarios.Abres roles no carga tabla
1,2,22,29,30	29	Accedes a usuarios. Abres roles, carga tabla y la muestra
1,2,22,29,30,31	30	Accedes a usuarios. Abres roles, carga tabla y la muestra, además deja

		registrar
1,2,22,29,30,31,32	31	Accedes usuarios. Abres roles, carga tabla y la muestra, además deja registrar y recarga la tabla
1,2,22,29,30,33	32	Accedes usuarios. Abres roles, carga tabla y la muestra, además deja eliminar y recarga la tabla
1,2,22,29,30,34	33	Accedes usuarios. Abres roles, carga tabla y la muestra, además deja modifica y recarga la tabla

Caja negra:
clases de equivalencia

Condición externa	Clases de Equivalencia Válidas	Clases de Equivalencia No Válidas
Está Registrado	ObtenerUsuario != null	ObtenerUsuario == null
Módulo eje Principal funciona	ObtenerEjePrincipal != nul RegistrarEjePrincipal != false EliminarEjePrincipal != false ModificarEjePrincipal != false	ObtenerEjePrincipal == nul RegistrarEjePrincipal == false EliminarEjePrincipal == false ModificarEjePrincipal == false
Módulo Geografía funciona	ObtenerGeografia != nul EliminarGeografia != false RegistrarGeografia != false ModificarGeografia != false	ObtenerGeografia ?= nul EliminarGeografia == false ModificarGeografia == false RegistrarGeografia == false
Módulo Perfiles funciona	ObtenerPerfiles != nul EliminarPerfiles != false ModificarPerfiles!= false RegistrarPerfiles !=false	ObtenerPerfiles == nul EliminarPerfiles == false ModificarPerfiles== false RegistrarPerfiles == false

Módulo Roles funciona	ObtenerRoles != nul EliminarRoles != false ModificarRoles!= false RegistrarRoles != false	ObtenerRoles == nul EliminarRoles == false ModificarRoles== false RegistrarRoles==false
Módulo Usuarios funciona	ObtenerUsuarios != nul EliminarUsuarios != false ModificarUsuarios!= false RegistrarUsuarios!=false	ObtenerUsuarios == nul EliminarUsuarios == false ModificarUsuarios== false RegistrarUsuarios ==false

6. Despliegue del proyecto.

Primero voy a explicar la visión que tendría un usuario al iniciar a la aplicación. Primero se encontraría con una página de Acceso principal. Donde el usuario ingresará mediante un nombre de usuario y contraseña. Este usuario y contraseña se deben de encontrar ya registrados para poder acceder. Una vez dentro se podrá ingresar uno nuevo o modificar el ya existente.

Posterior al ingreso encontraría la pantalla principal, donde el usuario visualiza distintos elementos como un dashboard y una tabla vertical en el lado derecho de la interfaz. La tabla muestra los distintos módulos: informes, organización, usuarios y encuestas. En los informes se muestran datos de las encuestas en gráficas que pueden ser diagrama de barras, columnas, sectores o de línea. El módulo Organización cuenta con apartados propios: eje organizativo, geografía y perfiles. El primero, que se llama, eje organizativo muestra todas las instalaciones que tiene la empresa/compañía en una tabla. Está relacionado con Geografía, que es una tabla estándar que vendría precargada en la aplicación para luego poder asignarla a la ubicación de las tiendas. Por último tenemos Perfiles, que sirve para mostrar la visibilidad de las preguntas/encuestas en las entidades. El siguiente se trataría de usuarios y se compone de las clases Usuarios y Roles. Cada uno con sus tablas propias y sus datos almacenados. En ellas se almacenan los usuarios registrados y sus roles dentro de la empresa. En último lugar tenemos el llamado Encuestas donde estarán incluidos: Campañas, Indicadores/Preguntas, Unidades y Tipos. Encuestas o Campañas, está formado por preguntas que se despliegan en las tiendas. Tienen un periodo de validez para poder

incluir datos en ellas, y pasado ese límite no se puede ingresar ningún dato. Seguido a Encuestas y relacionado a esta, tenemos la sección del módulo llamado Preguntas, que está formado por el conjunto de preguntas que forman una encuesta que deberá ser respondida para llegar a ser parte de las gráficas. Las siguientes secciones del módulo son; Unidades, donde se va a determinar el tipo de medida de la pregunta. Tipo, en cual si la pregunta es de entrada, es decir, que lo rellena la persona contribuidora.

Ahora después de describir la parte visible de la página web. Hablaré sobre el modelo arquitectónico usado, en nuestro caso un MVC, cómo funciona el programa internamente y cómo se relaciona con las 3 capas: “*CapaModelo*”, “*EncuestasWeb*” y “*CapaDatos*”.

La parte del controlador, es la encargada de controlar la relación entre las vistas y modelos. Recogerá las peticiones que llegan desde la vista y que interactúan con los modelos. Consta de distintos controladores cada uno basado en un objeto de la “*CapaModelo*”, que en nuestro caso sería la parte modelo del MVC, pero que se encuentra fuera de “*EncuestasWeb*”, y debe ser importada.

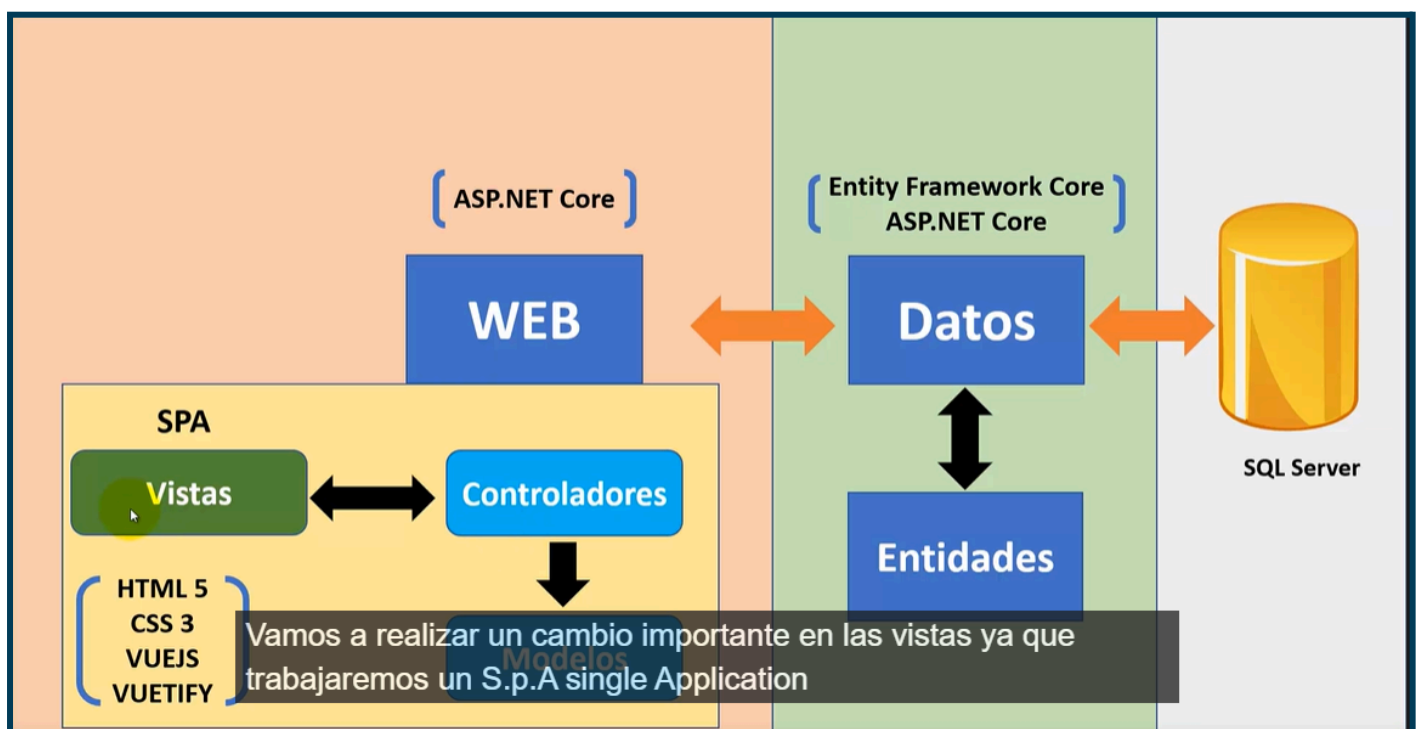
Tanto los controladores creados, como los métodos y objetos importados, se encuentran escritos en base al lenguaje c# de Visual Studio .NET. Describiendolo de mejor manera, los controladores tratan de archivos donde están programados los métodos que serán llamados desde la vista, estos utilizan métodos de la “*CapaDatos*”, hechos concretamente para poder conectarse a la BBDD e interactuar con ella. Claro está, que usamos objetos de la capa modelo que son necesarios para implementar en correcto funcionamiento los métodos propios. Podemos decir que la parte del controlador y modelo forman parte del back-office, que el cliente no verá.

La parte de vista recibe datos del modelo y los muestra para que pueda verlo el cliente. Programada en **cshtml**, siendo este un lenguaje concreto de ASP.NET, nos permitirá elaborar la página web, siendo parte del front-office y la única que el cliente verá en pantalla. Por lo tanto se trata de páginas web basadas en html, css y en el uso de scripts de javascript.

En la parte de vista, tenemos distintas views, desde la principal hasta views más específicas. La vista principal, llamada layout, se encarga conectar el front con el back, realizando una conexión con el controlador y recibiendo datos, que mandará por medio de ajax a ficheros javascript, lo cuales serán los encargados de complementar con los datos, recibidos en

formato json, sus view asociadas. Estas views asociadas serán los distintos módulos y páginas que forman parte de la aplicación web.

En el apartado de BBDD, como he mencionado anteriormente usamos SQL. Creando nuestra propia bbdd y sus respectivas tablas. Cada una de estas tablas y sus propias columnas tienen relación con su respectivo objeto, de igual nombre y ubicado en “*CapaModelo*”. Este objeto, cuenta con unos atributos del mismo tipo que las columnas de su tabla relacionada. Convertimos los datos almacenados en tablas, en objetos, para facilitar su uso. Además de la conversión de la base de datos a objetos c#, tenemos la creación de procedimientos.



7. Problemas encontrados.

Durante el desarrollo de nuestro proyecto pasamos por distintos problemas de complejidad variable. El primero, sería la falta de tiempo, quisimos abarcar mucho y tuvimos que ir desechando ideas, como por ejemplo los gráficos. Otras dificultades que tuvimos durante el proceso de producción, serían pequeños errores de sintaxis a la hora de transmitir datos vía ajax entre el index html y su javascript correspondiente.

8. Conclusiones. (Evaluación crítica de los resultados obtenidos. Propuestas de mejoras y desarrollos futuros).

Para no haber hecho nunca un proyecto parecido, creemos que no ha salido mal, además hemos aprendido un montón acerca del funcionamiento de MVC y la transmisión de datos entre controladores, vistas y modelos. En cuanto a mejoras y posibles desarrollos sería la introducción de distintas gráficas para acompañar las tablas y hacer del análisis de datos una tarea más fácil de realizar, ya que solo tenemos una gráfica, nos gustaría más variedad en cuanto a estas.

9. Bibliografía.

<https://www.youtube.com/watch?v=-dO2QG3OQzE&t=353s>

<https://findstack.com/resources/cloud-computing-statistics/>

<http://www.lsi.us.es/docencia/get.php?id=361>

<https://www.tecnologias-informacion.com/modelos-datos.html>

<https://bookdown.org/paranedagarcia/database/modelamiento-de-datos.html#modelamiento-f%C3%ADsico>

<https://normasyestandaresproyectosti.wordpress.com/2015/01/29/iso-12207/>

http://fcaenlinea.unam.mx/anexos/1728/Unidad_2/u2_act2_1.pdf