Descripción del sistema

Corporación Juntos Construyendo Futuro



Presentado por:

Fabian Andres Olarte Varga

**Tabla de contenidos.**

1. Introducción.
2. Aspectos iniciales del proyecto.
   1. Visión del software.
   2. Objetivos.
   3. Restricciones.
3. Entregables y calendarización.
   1. Evolución del plan.
4. Glosario.
5. Contexto del proyecto.
   1. Modelo de ciclo de vida.
   2. Lenguajes y herramientas.
6. Organización del proyecto y comunicación.
   1. Interfaces externas.
   2. Organigrama y descripción de roles.
7. Funciones del producto.
   1. Funcionalidades.
   2. Caracteristicas del usuario.
   3. Requisitos no funcionales.
8. Plan de trabajo.
   1. Mapa de historias de usuario.
   2. Herramientas de estimación.
9. Arquitectura.
   1. Vista lógica del sistema.
      1. Diagrama de componentes.
      2. Diagrama de paquetes.
   2. Vista física del sistema.
      1. Diagrama de despliegue.
   3. Diseño detallado.
      1. Diagrama de clases.
      2. Persistencia
   4. Interfaz de usuario.
10. Anexos.

**Lista de figuras.**

* Figura 1 – Modelo Canvas
* Figura 2. Ciclo de vida del sistema
* Figura 3.
* Figura 4. Diagrama de componentes
* Figura 5. Diagrama de paquetes
* Figura 6. Diagrama de despliegue
* Figura 7. Diagrama de clases
* Figura 8. Diagrama de persistencia

**Lista de tablas.**

* Tabla 1 – Entregables a lo largo del proyecto.
* Tabla 2. Organización del equipo.
* Tabla 3. Funcionalidad 1
* Tabla 4. Funcionalidad 2
* Tabla 5. Funcionalidad 3
* Tabla 6. Funcionalidad 4
* Tabla 7. Funcionalidad 5
* Tabla 8. Funcionalidad 6
* Tabla 9. Funcionalidad 7
* Tabla 10. Funcionalidad 8
* Tabla 11. Funcionalidad 9
* Tabla 12. Funcionalidad 10
* Tabla 13. Funcionalidad 11
* Tabla 14. Funcionalidad 12
* Tabla 15. Funcionalidad 13
* Tabla 16. Funcionalidad 14
* Tabla 17. Funcionalidad 15
* Tabla 18. Funcionalidad 16
* Tabla 19. Funcionalidad 17
* Tabla 20. Caracteristicas Profesional de campo
* Tabla 21. Caracteristicas Coordinador
* Tabla 22. Caracteristicas Directivo
* Tabla 23. Caracteristicas Administrador
* Tabla 24. Caracteristicas Super usuario
* Tabla 25. Estimación.

1. **Introducción.**

El presente documento tiene como propósito establecer las bases del proyecto, en este se fijarán las herramientas que se utilizarán para el desarrollo de este, como serán los roles de cada integrante del equipo en la implementación de este proyecto, las funcionalidades que tendrá y de cómo será el diseño detallado de este software

La propuesta planteada se basa en un software a modo de sistema de informacion capaz de gestionar información que es recolectada en cada uno de los procesos de la *Corporación Juntos Construyendo Futuro,* tambien ayudara a facilitar la visualización del estado de cada uno de los proyectos desarrollados por la corporación, es decir, dar un seguimiento del proceso de cada uno de los proyectos, más adelante profundizaremos en cada una de estas funcionalidades.

Un vez finalizado el documento se tiene como objetivo facilitar el entendimiento de cómo será el desarrollo de este proyecto, ya que al ser a mediano y largo plazo se necesitará ser específicos para que cualquier equipo pueda entender y a su vez continuar desarrollando sin necesidad de volver a pasos ya realizados anteriormente.

1. **Aspectos iniciales del proyecto**

**2.1 Visión del software.**

El software a desarrollar tiene como objetivo facilitar y optimizar procesos de trabajo de la *Corporación Juntos Construyendo Futuro,* este software será capaz de facilitar la gestión y la recopilación de los diferentes escenarios de trabajo, puesto que uno de sus procesos de trabajo es realizar visitas a diferentes comunidades a lo largo del territorio nacional, por eso se necesitará de una herramienta capaz de recopilar la informacion en cualquier lugar y tener a su vez una gestión centralizada de toda esta informacion recolectada. Por otra parte este software tambien brindará un control y seguimiento en cada proyecto, es decir, puesto que la corporación realiza diferentes proyectos a mediano y largo plazo se necesita una herramienta que pueda ayudar al seguimiento y control del estado de cada uno de estos proyectos. Y por último brindara un espacio para consultar y analizar toda esta información recolectada para llevar a cabo de forma completa y ordenada cada proyecto.

Este software dispondrá de varios usuarios los cuales tendrán diferentes roles, por lo cual se desarrollará un control de las funcionalidades y acceso a datos que pueda usar cada usuario, todo esto con el fin de que cada rol pueda cumplir su labor especifica y haya un orden en cada proceso. Los diferentes tipos de usuarios serán: *Profesional de campo* el cual es encargado de movilizarse a diferentes comunidades y realizar diferentes acompañamientos a lo largo del tiempo, este tipo de usuario tendrá la disponibilidad de recopilar y cargar información al software desde cualquier parte, además tendrá acceso a su historial de informes cargados y datos de los integrantes de las comunidades que estén trabajando. Luego encontramos a los *Coordinadores* los cuales son encargados de dirigir equipos conformados de profesionales de campo, con el fin de seguir y controlar el proceso a llevar para el desarrollo de un proyecto y/o tarea especifica, es por eso por lo que serán capaces de fijar un plan de acción (lista de tareas pendientes) para llevar a cabo los proyectos, tambien podrán enviar retroalimentación a la información cargada de los profesionales de campo y validar si se puede cargar dicha informacion o necesita ser corregida, además contarán con una ayuda que les permita saber si los profesionales de campo han cumplido con sus acompañamiento y por último el coordinador tendrá acceso a toda la informacion cargada por los profesionales de campo. El siguiente tipo de usuario es el *directivo* el cual se encarga de visualizar el proceso y tomar acciones dependiendo al proceso actual para poder llevar a cabo con éxito los diferentes proyectos, este tendrá acceso a toda la informacion detallada del proyecto con el fin de hacer seguimientos. Tambien tenemos al *administrador* del sistema el cual podrá personalizar variables de estado, formularios y reportes de carga, es decir, se encarga de gestionar la parte especifica de la administración de un proyecto, asignando coordinador y los profesionales de campo, y por último tenemos al *super usuario* el cual podrá gestionar los roles, es decir, podrá asignar diferentes roles a cada usuario, tambien puede crear módulos para la gestión correcta de un proyecto, por ejemplo, crear módulos de carga de reportes, módulos de validación, etc. Además tendrá acceso a todos los datos existentes en el sistema para hacer validaciones de calidad y por último tambien será capaz de cumplir el mismo rol del administrador. Cabe resaltar que los usuarios podrán estar en varios proyectos al mismo tiempo, por lo cual es necesario que la herramienta brinde un espacio donde se encuentren todos los proyectos a los que el usuario este asociado, en el caso del super usuario como se mencionó antes, tendrá acceso a todos los proyectos existentes en la plataforma.

Timeline

Description automatically generated

**Figura 1 – Canvas**

En la *figura 1* se puede evidenciar el modelo Canvas para un entendimiento del producto, cabe resaltar que toda la informacion que se está mostrando en las historias de usuario disponibles en los recursos al final del documento.

**2.2 Objetivos.**

**Objetivo general:** Desarrollar un software para gestionar y recopilar informacion, además que pueda proporcionar una ayuda en la visualización del estado de avance de cada uno de los proyectos a desarrollar.

**Objetivos específicos:**

* El software será capaz de ayudar en la recopilación de datos y poder cargarlos desde cualquier parte.
* Brindar un espacio de ayuda en el seguimiento y control del avance de cada proyecto.
* Ayudar en el proceso de búsqueda de informacion mediante filtros para facilitar su búsqueda.
* Dar un control de versiones de cada formulario que se suba a este (transversalidad).
* Ayudar en el proceso de llenado de la información.
* Etiquetar la informacion.
* Tener la informacion de cada miembro de las diferentes comunidades trabajadas.
* Brindar una estructura de variables (variables obligatorias y opcionales en formularios) para la recopilación ordenada de datos.
* Ayudar con la creación de indicadores operativos o de estado para visualizar el avance de cada proyecto.
* Brindar un espacio donde se encuentren las tareas pendientes para cada persona.
* Dar la posibilidad de crear módulos para poder administrar de una mejor forma un proyecto.
* Enviar correo de confirmación al usuario cuando la información sea cargada con éxito.
* Poder agregar miembros de las diferentes comunidades que se vayan trabajando para tener su informacion disponible cuando se necesite.
* Tener un espacio para que se pueda validar la informacion antes de ser cargada al sistema.

**2.3 Restricciones.**

Una vez desarrollado el software se necesitará el recurso de una base de datos la cual será capaz de guardar toda la informacion necesaria para el correcto funcionamiento de este. Además se necesitará contar con un servidor el cual se va a encargar de tener funcionando el software.

1. **Entregables – Calendarización**

|  |  |  |
| --- | --- | --- |
| Numero de entregable | Descripción de entregable | Fecha |
| 1 | Recolección de la informacion: Dirigir sesiones con diferentes personas con el fin de recoger los requisitos que debe tener el software. | 08/03/2022 |
| 2 | Descripción del sistema a desarrollar: En base a la información recogida se desarrollará un documento explicando formalmente el propósito y los objetivos que deberá tener el sistema. | 15/03/2022 |
| 2.1 | Descripción técnica del desarrollo del producto: Se establecerán los parámetros para el desarrollo de del sistema, tales como el ciclo de vida que tendrá este, lenguajes y herramientas a utilizar, la organización del proyecto, funcionalidades y plan de trabajo del proyecto. | 12/04/2022 |
| 3 | Diseño de los procesos y comportamiento del sistema: Se diseña los procesos que estructuran el sistema, además se prestara la vista lógica del sistema y muckups. | 27/04/2022 |
| 3.1 | Diseño de las interfaces: Se diseñarán las interfaces en las herramientas y lenguajes descritos. | --- |

**Tabla 1 – Entregables a lo largo del proyecto**

En la anterior tabla se puede evidenciar como serán los entregables a los largo del proyecto, cabe resaltar que al ser un proyecto a mediano y largo plazo varios equipos estarán trabajando en este, por lo cual se requiere que cada equipo agregue sus entregables de acuerdo con el estado del proyecto y de acuerdo con el tiempo que estén manejando.

**3.1 Evolución del plan.**

Como se ha venido mencionando este proyecto se desarrollará a mediano y largo plazo por lo cual en esta sección será importante destacar cuales son las secciones que variaran con el paso de tiempo y con el paso de los diferentes equipos que pasen por el proyecto. Las secciones que van a variar son:

* **Entregables – Calendarización:** La sección de entregables – calendarización debe ser actualizada por los diferentes equipos y con el alcance que ellos le den al tiempo que van a estar desarrollando el proyecto.
* **Evolución del plan:** Esta misma sección se deberá ir completando, dependiendo el alcance del trabajo que vaya a desarrollar el equipo.
* **Glosario:** La sección de glosario debe estar siempre actualizada para que el lector pueda entender la totalidad del documento.
* **Organización del proyecto y comunicación:** La sección de organización del proyecto y comunicación va a variar dependiendo los roles que van a estar dentro del equipo. Es importante tener actualizado el contacto de la entidad externa ya que puede variar a lo largo del proyecto, por otra parte se debe diligenciar el organigrama y la tabla describiendo los roles de cada persona y responsabilidades que tendrá cada uno.
* **Funcionalidades:** En esta sección el equipo encargado debe proporcionar la informacion de horas estimadas y responsable de implementación a cada funcionalidad establecida, cabe resaltar que las horas estimadas que se deben poner son el resultado de lo que el equipo de implementación debe realizar en las herramientas de estimación.
* **Plan de trabajo del proyecto:** En esta sección es indispensable que el equipo que se encargue de desarrollar el diseño planteado, diligencia como será el plan de trabajo para desarrollar el software. En esta sección estará la estimación de las tareas y como se va a organizar la implementación de estas.
* **Arquitectura:** Esta sección será pertinente que el equipo entienda cada diagrama y en caso de tener propuesta de mejora se debe realizar este cambio al diagrama, para una vez finalizado se pueda iniciar con la implementación del producto. Tambien en dado caso de tener propuestas de mejor para el diseño de la herramienta será pertinente cambiar el diseño por el acordado con la organización.

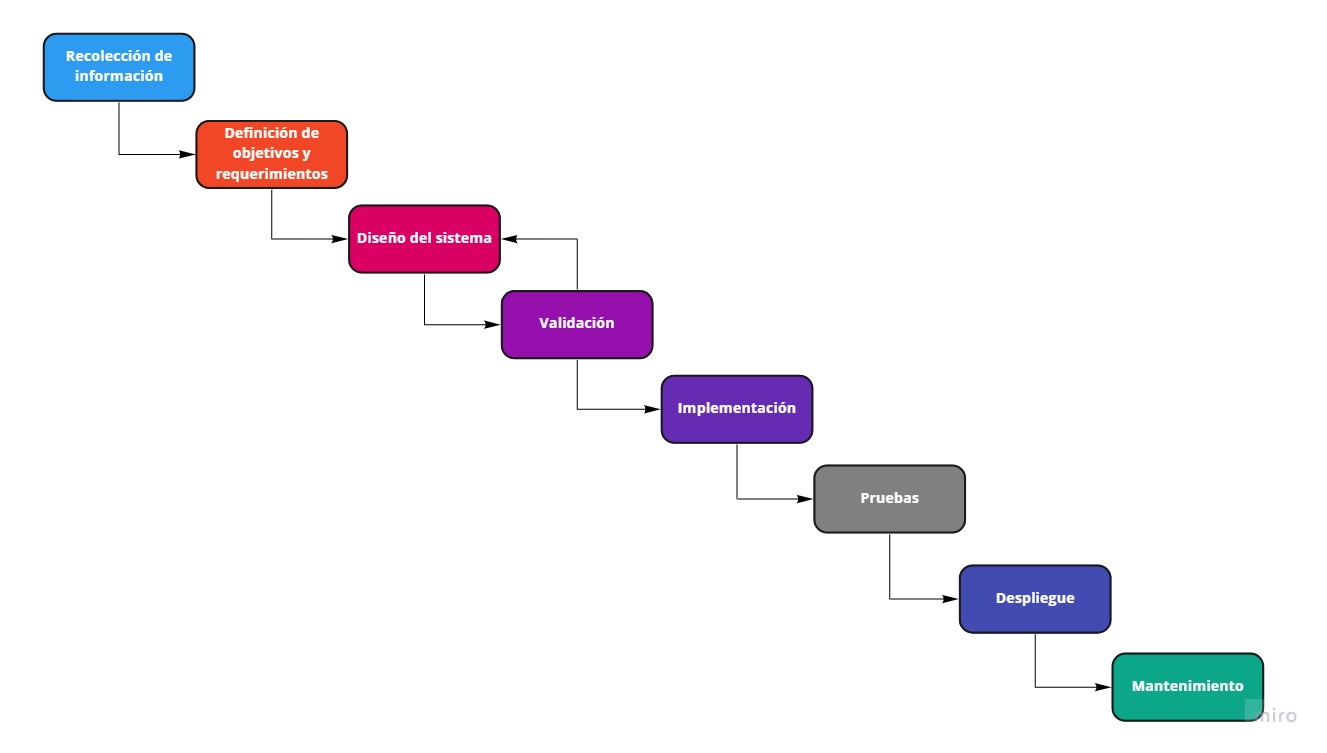
1. **Glosario.**

En esta sección se explicará el significado de algunas palabras que se consideren importantes para un correcto entendimiento, dicho esto a continuación las palabras:

* **Indicadores operativos:** Son indicadores de avance que describen el estado de un proyecto.
* **Plan de accion:** Lista de tareas que se deben cumplir para desarrollar un proyecto y/o tarea especifica.
* **Variables de información:** Tipo de informacion que debe ser cargada, puede ser opcional u obligatoria.
* **Beneficiario:** Persona que fue beneficiaria de un proyecto, o personas acompañadas.

1. **Contexto del proyecto.**

**5.1 Modelo de ciclo de vida del proyecto.**

****

**Figura 2. Ciclo de vida del sistema**

Partiendo del hecho de que este el desarrollo de este sistema va a ser a mediano y largo plazo y que por ende varios equipos de trabajo van a pasar a desarrollarlo, se propuso un ciclo de vida en cascada con una pequeña variación la cual consiste en una validación después del diseño del sistema, ya que cada equipo dependiendo a sus habilidades y su entendimiento del problema se pueden presentar algunos cambios en los parámetros de la planeación y entregables de su esquema de trabajo. Cabe resaltar que esta validación tambien va implícita en la implementación puesto que se va a trabajar con una metodología scrum y es necesario que se planteen entregables para validarlos con el cliente que en este caso es la corporación juntos construyendo futuro. Por último cabe resaltar la importancia de que cada equipo que retome el desarrollo inicie en la etapa de definición de objetivos y requerimientos, y diseño del sistema puesto que es donde se ve evidenciada la explicación como será el software, para que el equipo pueda comprender el trabajo que se lleva actualmente para que posteriormente puedan realizar sus modificaciones y puedan completar algunas partes como estimación, organización del equipo, entregables dependiendo de su esquema de trabajo, para que posteriormente puedan validarlo con la corporación y asi se pueda realizar una correcta implementación.

**5.2 Lenguajes y herramientas.**

Para la elaboración de este software se propone una aplicación web que sea capaz de ejecutarse tanto en dispositivos móviles como en computadores, todo con el fin de realizar de forma mas sencilla este software, considerando este factor a continuación se describen los lenguajes:

**Framework:**

Angular.

Angular es un framework opensource desarrollado por la empresa Google, el cual tiene como objetivo facilitar la creación y programación de aplicaciones web de una sola página. Se escogió este framework ya que, como se mencionó anteriormente, se buscaba facilidad, agilidad y eficacia a la hora de desarrollar tanto el front-end como el back-end de la aplicación web, y angular encaja perfectamente, pues una de las ventajas más destacadas que tiene este framework es que separa completamente estos 2 apartados evitando así código repetitivo, esto lo consigue gracias a su patrón MVC (Modelo-Vista-Controlador) y asegura el desarrollo ágil, eficaz y ordenado del código de la aplicación web.

**FRONT-END:**

*HTML.*

HTML es un lenguaje de marcado usado esencialmente para el desarrollo de páginas web, el cual nos ayuda a indicar como es el orden del contenido del front-end de la página web mediante el uso de etiquetas o *tags*. Se decidió usar HTML puesto que, además de ser uno de los lenguajes de marcado web más populares, se complementa muy bien con otras 2 herramientas que se usarán para el desarrollo del front-end de la aplicación que ya se explicarán más adelante en sus respectivos apartados y las cuales son CSS y Typescript.

*CSS.*

CSS es un lenguaje de marcado, también conocido por ser la “hoja de estilos” de la página web, y está enfocado en definir, crear y mejorar la presentación de una página y/o aplicación web. Se hará uso de esta herramienta pues trabaja muy bien junto a HTML para la realización del front-end de la aplicación web, ya que CSS se encarga de darle una personalización al aspecto que tendrá la estructura u organización ya planteada por HTML.

*Typescript.*

Typesrcipt es un superset de JavaScript, esto significa que los programas de JavaScript también son válidos para Typescript, sin importar que sean lenguajes de programación diferentes. Se decidió hacer uso de Typesript para complementar la fase de desarrollo del front-end puesto que, aparte de que trabaja de forma óptima junto con CSS y HTML, es el lenguaje usado por Angular para el desarrollo del front-end de sus aplicaciones web, pudiendo de esta forma usar sus librerías y trabajar de forma cómoda en este framework.

*Angular Material.*

Angular Material es una librería de estilos para el desarrollo del “front” de aplicaciones web, está basada en la guía de diseño de Material Design, y está desarrollada por el equipo de Angular. Se decidió hacer uso de estas librerías ya que son bastante completas, lo cual ayuda a que la estética de la aplicación sea agradable e intuitiva, además, al ser desarrolladas por el equipo de Angular, encajan perfectamente con todas las herramientas ya mencionadas.

**BACK-END:**

*JavaScript.*

JavaScript es un lenguaje de programación que permite implementar funciones complejas en el desarrollo de una página o aplicación web. Se quiso hacer uso de JavaScript para el back-end de la aplicación pues es la opción más adecuada para ello, como ya se mencionó anteriormente es un lenguaje de programación que nos ayudara con la funcionalidad de la página sin preocuparse por la estética de esta.

*Mongoose.*

Mongoose es una biblioteca de JavaScript que permite definir esquemas con datos fuertemente tipados. Esta biblioteca permite crear un modelo basado en el esquema especificado anteriormente, dicho modelo se acopla a un documento MongoDB, el cual permite la realización de distintas funciones como validar, guardar, eliminar y/o consultar los datos previamente descritos. Se quiso hacer uso de esta biblioteca ya que complementa de forma óptima y eficaz al lenguaje JavaScript, ayudando así al desarrollo del back-end de la aplicación.

1. **Organización del proyecto y comunicación.**

**6.1 Interfaces externas.**

La entidad externa de este proyecto será la *corporación juntos construyendo futuro*, la cual es la será beneficiada con este desarrollo, la forma de comunicarse con esta entidad es a través de la materia *proyecto social universitario*, la cual pondrá en contacto a los desarrolladores con el interlocutor de esta entidad, el actual contacto para comunicarse con la entidad es: [diegopuerta@juntosconstruyendofuturo.org](mailto:diegopuerta@juntosconstruyendofuturo.org), cabe resaltar que este correo puede estar sujeto a cambios dependiendo la corporación.

**6.2 Organigrama y descripción de roles.**

Esta sección deberá ser desarrollada por el equipo que vaya a implementar el software, acá será necesario poner un organigrama en el cual se evidencia el rol que va a tener cada miembro del equipo, es recomendable trabajar con la metodología scrum y en este orden de ideas se deberá tener los roles de product owner que en este caso será la corporación, el scrum master y por último la distribución que tendrá el equipo a lo largo del desarrollo, por ejemplo, alguien estará a cargo del front end y otro del back end.

Una vez realizado el organigrama se deberá diligenciar la siguiente tabla la cual tendrá la explicación de cada rol que va a cumplir cada persona.

|  |  |  |  |
| --- | --- | --- | --- |
| **Rol** | **Persona a cargo** | **Descripción del rol** | **Responsabilidades** |
| Product owner |  |  |  |
| Scrum master |  |  |  |
| Scrum team |  |  |  |
| … |  |  |  |
| … |  |  |  |

Tabla 2. Organización del equipo.

En la tabla se debe diligenciar que responsabilidades tendrá cada rol y a su vez que persona tomará el rol, además se debe describir que debe hacer el rol, cabe resaltar que la tabla puede variar dependiendo el equipo y como este se vaya a distribuir sus tareas.

1. **Funciones del producto.**

**7.1 Funcionalidades.**

En esta sección se documentarán las funcionalidades en base a las historias de usuario.

|  |  |
| --- | --- |
| Nombre de la historia: Transversalidad en los datos | |
| Numero: 1 | Usuario: Coordinador |
| Prioridad en negocio: 3 | Horas estimadas: |
| Programador responsable: | |
| Descripción: Como coordinador quiero que exista transversalidad entre los datos que permita hacer análisis para la construcción de informes mensuales más eficientes. | |
| Validación: El software debe mostrar un historial de versiones de la información. | |

Tabla 3. Funcionalidad 1

|  |  |
| --- | --- |
| Nombre de la historia: Espacio centralizado donde se va a encontrar toda la información. | |
| Numero: 2 | Usuario: Coordinador y profesional de campo |
| Prioridad en negocio: 5 | Horas estimadas: |
| Programador responsable: | |
| Descripción: Como coordinador quiero poder encontrar la información en un mismo espacio para evitar la manipulación de los mismos datos en distintas herramientas. | |
| Validación: El software debe ser de facil manejo, donde este consolidada la informacion de cada proyecto, además los usuarios (coordinador y profesional de campo) dependiendo al proyecto solo podrán acceder a la información de ese mismo. Se debe asegurar la atomicidad en los datos para que no haya repeticiones. | |

Tabla 4. Funcionalidad 2

|  |  |
| --- | --- |
| Nombre de la historia: Etiquetado de la información. | |
| Numero: 3 | Usuario: Coordinador y profesional de campo. |
| Prioridad en negocio: 3 | Horas estimadas: |
| Programador responsable: | |
| Descripción: Como coordinador me gustaría buscar la información a través de etiquetas para no tener que buscar uno por uno y asi poder agilizar el trabajo. | |
| Validación: El profesional de campo debe ser capaz de etiquetar la informacion cuando la vaya a subir al sistema, además el coordinador debe ser capaz de buscar la informacion mediante esas etiquetas. | |

Tabla 5. Funcionalidad 3

|  |  |
| --- | --- |
| Nombre de la historia: Historial de las personas. | |
| Numero: 4 | Usuario: Profesional de campo. |
| Prioridad en negocio: 3 | Horas estimadas: |
| Programador responsable: | |
| Descripción: Como profesional de acompañamiento me gustaría poder tener una plataforma que me permita acceder al "historial" de cada joven. Para poder conocer con mayor facilidad la información de cada joven. | |
| Validación: El profesional de campo debe poder ver el historial de las personas que estan siendo beneficiadas, es decir, se brindara un espacio en donde este toda la información, además se debe poder buscar por filtros. El acceso a esta información dependerá en el proyecto el cual este asociado el profesional de campo. Tambien se debe poder eliminar a personas (beneficiaros del proyecto) si ya no estan asociadas al proyecto. | |

Tabla 6. Funcionalidad 4

|  |  |
| --- | --- |
| Nombre de la historia: Descargar reportes de la plataforma. | |
| Numero: 5 | Usuario: Coordinador. |
| Prioridad en negocio: 2 | Horas estimadas: |
| Programador responsable: | |
| Descripción: Como coordinador, quiero poder descargar reportes para facilitar su revisión cuando no tenga conectividad. | |
| Validación: Aparte de poder visualizar todos los reportes, se debe poder tener la opción de descargar cualquiera de estos. | |

Tabla 7. Funcionalidad 5

|  |  |
| --- | --- |
| Nombre de la historia: Subida de informacion a la plataforma. | |
| Numero: 6 | Usuario: Profesional de campo. |
| Prioridad en negocio: 5 | Horas estimadas: |
| Programador responsable: | |
| Descripción: Como profesional de campo quiero poder subir la información (informes) desde cualquier parte (lugar). | |
| Validación: El profesional de campo tendrá un espacio de llenado de informacion recogida para posteriormente, poder subirla a la plataforma desde cualquier lugar. | |

Tabla 8. Funcionalidad 6

|  |  |
| --- | --- |
| Nombre de la historia: Validación y ayuda de llenado de información. | |
| Numero: 7 | Usuario: Profesional de campo. |
| Prioridad en negocio: 5 | Horas estimadas: |
| Programador responsable: | |
| Descripción: como profesional de campo quiero que al cargar la información hubiera identificación automática de los datos para evitar errores de digitación.  Como profesional de campo quiero reportar la información en una herramienta que permita cargar los datos de manera ágil para evitar pérdidas de tiempo en diligenciamientos básicos.  Como profesional de campo me gustaría tener mi informacion automáticamente para tenerla disponible al momento de llenar un formulario.  Como profesional de campo me gustaría tener toda la informacion personal de cada joven de forma automática para no tener que llenar los formularios con la misma información. | |
| Validación: Al momento de que el profesional de campo vaya a subir la informacion, se debe validar si todos los campos estan correctos, además debe haber opción de autocompletar cada campo (formulario o reporte) con la información básica del beneficiario en caso de existir, y ademas la información básica del profesional de campo que este diligenciando el reporte. | |

Tabla 9. Funcionalidad 7

|  |  |
| --- | --- |
| Nombre de la historia: Verificación de subida de la información a la plataforma. | |
| Numero: 8 | Usuario: Profesional de campo |
| Prioridad en negocio: 5 | Horas estimadas: |
| Programador responsable: | |
| Descripción: Como profesional de campo deseo recibir al correo electrónico el PDF de las herramientas (tipos de formularios o reportes) aplicadas. | |
| Validación: Se debe enviar un correo electrónico de verificación al profesional de campo, además en este correo debe estar el reporte enviado. | |

Tabla 10. Funcionalidad 8

|  |  |
| --- | --- |
| Nombre de la historia: Indicadores operativos. | |
| Numero: 9 | Usuario: Coordinador. |
| Prioridad en negocio: 5 | Horas estimadas: |
| Programador responsable: | |
| Descripción: Como coordinador quiero tener control y acceso a los indicadores operativos (indicadores avance) del proyecto para hacer seguimiento más eficiente.  Como coordinador quiero poder ingresar las acciones del plan de acción del equipo para poder hacer seguimiento a los indicadores de resultado. | |
| Validación: El coordinador debe ser capaz de crear indicadores operativos en el proyecto, además debe ser capaz de asignar tareas a los profesionales de campo. | |

Tabla 11. Funcionalidad 9

|  |  |
| --- | --- |
| Nombre de la historia: Retroalimentación o solicitudes de corrección. | |
| Numero: 10 | Usuario: Coordinador. |
| Prioridad en negocio: 4 | Horas estimadas: |
| Programador responsable: | |
| Descripción: Como coordinador, quiero poder enviar retroalimentaciones o solicitudes de corrección al equipo manejando una sola plataforma para garantizar que todos los ajustes queden validados.  Como coordinador territorial quiero tener acceso a la información para validarla y hacer constante seguimiento a la información cargada | |
| Validación: Cuando el profesional de campo sube la informacion debe pasar por un proceso de validación en donde el coordinador puede aceptar que se suba la informacion o enviar petición de corrección si se debe corregir algo. | |

Tabla 12. Funcionalidad 10

|  |  |
| --- | --- |
| Nombre de la historia: Verificar el cumplimiento de los profesionales de campo. | |
| Numero: 11 | Usuario: Coordinador. |
| Prioridad en negocio: 4 | Horas estimadas: |
| Programador responsable: | |
| Descripción: Como coordinadora quiero contar con una herramienta que vaya consolidando las fechas de los acompañamientos, permite identificar que se tienen los 2 acompañamientos y aprobar pagos (ver lo que hizo cada persona).  Como coordinador quiero tener visualización del avance en el cargue de información solicitada para hacer seguimiento al equipo en tiempo real. | |
| Validación: El coordinador aparte de añadir tareas a profesionales de campo, tambien podrá visualizar el avance de trabajo de cada profesional de campo. | |

Tabla 13. Funcionalidad 11

|  |  |
| --- | --- |
| Nombre de la historia: Apartado de tareas pendientes. | |
| Numero: 12 | Usuario: Coordinador y profesional de campo. |
| Prioridad en negocio: 4 | Horas estimadas: |
| Programador responsable: | |
| Descripción: Como profesional de campo y coordinador me gustaría tener un apartado de tareas pendientes para tener más control en mi tiempo (fecha de vencimiento y recuerdos con alertas).  Como profesional quiero ver mi cumplimiento de tareas para evitar subir varias veces la misma informacion | |
| Validación: Los usuarios tendrán un apartado de tareas pendientes para que tengan un orden en su trabajo. El coordinador podrá asignar tareas a los profesionales de campo. | |

Tabla 14. Funcionalidad 12

|  |  |
| --- | --- |
| Nombre de la historia: Gestion de permisos. | |
| Numero: 13 | Usuario: Super usuario. |
| Prioridad en negocio: 5 | Horas estimadas: |
| Programador responsable: | |
| Descripción: Como super usuario quiero poder crear perfiles y definir los permisos para tener control sobre quien gestiona la plataforma. | |
| Validación: El super usuario podrá asignar roles a cada usuario registrado. | |

Tabla 15. Funcionalidad 13

|  |  |
| --- | --- |
| Nombre de la historia: Gestion especifica de cada proyecto. | |
| Numero: 14 | Usuario: Administrador y super usuario. |
| Prioridad en negocio: 5 | Horas estimadas: |
| Programador responsable: | |
| Descripción: Como administrador quiero poder personalizar las variables, formularios y reportes de carga. | |
| Validación: El administrador tendrá un espacio para crear proyectos y a su vez añadir formularios, reportes y dentro de estas añadir variables de información para cargar la información. | |

Tabla 16. Funcionalidad 14

|  |  |
| --- | --- |
| Nombre de la historia: Reportes para directivos. | |
| Numero: 15 | Usuario: Super usuario. |
| Prioridad en negocio: 5 | Horas estimadas: |
| Programador responsable: | |
| Descripción: Como super usuario quiero poder personalizar los reportes para los Directivos. | |
| Validación: El super usuario tendrá un módulo para modelar un reporte con informacion como estadísticas que se vayan a mostrar a los directivos. | |

Tabla 17. Funcionalidad 15

|  |  |
| --- | --- |
| Nombre de la historia: Acceso a toda la información. | |
| Numero: 16 | Usuario: Super usuario. |
| Prioridad en negocio: 4 | Horas estimadas: |
| Programador responsable: | |
| Descripción: Como super usuario quiero poder tener acceso a todos los módulos del sistema para su gestión y soporte a los otros perfiles  Como super usuario quiero tener acceso a la data del sistema sin restricción para hacer validaciones de calidad de la información. | |
| Validación: El super usuario podrá acceder a la informacion de cada proyecto, además de poder modificar cosas de los proyectos. | |

Tabla 18. Funcionalidad 16

|  |  |
| --- | --- |
| Nombre de la historia: Trazabilidad de herramientas utilizadas. | |
| Numero: 17 | Usuario: Coordinador y profesional de campo. |
| Prioridad en negocio: 3 | Horas estimadas: |
| Programador responsable: | |
| Descripción: Como profesional y como coordinadora, contar con la trazabilidad de las herramientas psicosociales que se aplican en todas las etapas del acompañamiento | |
| Validación: El coordinador y el profesional de campo podrá visualizar el historial de las herramientas aplicadas. | |

Tabla 19. Funcionalidad 17

**7.2 Caracteristicas del usuario.**

En esta sección se documentará las caracteristicas como nivel de privilegio, roles, nivel de experiencia técnica y frecuencia de uso que realizará cada usuario.

**Profesional de campo**

|  |  |
| --- | --- |
| **Característica del usuario** | **Descripción** |
| **Nivel de privilegios** | * Poder subir información a la plataforma. * Visualizar el historial de la información subida. * Poder visualizar las tareas pendientes que le son asignadas por el coordinador o por este mismo. |
| **Roles** | * Profesional de campo: Tiene la disponibilidad de subir información a la plataforma. |
| **Nivel de experiencia técnica** | Se necesita una experiencia muy básica en el manejo de herramientas virtuales. Ya que la plataforma será de facil manejo. |
| **Frecuencia de uso** | Cada vez que necesite realizar una tarea (acompañamientos). |

Tabla 20. Caracteristicas Profesional de campo

**Coordinador**

|  |  |
| --- | --- |
| **Característica del usuario** | **Descripción** |
| **Nivel de privilegios** | * Poder ver la información generada por los profesionales de campo. * Poder añadir indicadores operativos o planes de acción. * Poder asignar tareas a los profesionales de campo. * Poder ver el progreso de los proyecto. * Poder aceptar la información subida por los profesionales de campo, tambien en dado caso, poder enviar solicitudes de corrección o retroalimentación. |
| **Roles** | * Profesional de campo: Tiene la disponibilidad de subir información a la plataforma. * Coordinador: Tiene la disponibilidad de ver y gestionar el progreso de cada proyecto. |
| **Nivel de experiencia técnica** | Se necesita una experiencia media para el uso de los diferentes apartados que va a tener que coordinar. |
| **Frecuencia de uso** | Cada vez que necesita gestionar el proceso de cada proyecto. |

Tabla 21. Caracteristicas Coordinador

**Directivo**

|  |  |
| --- | --- |
| **Característica del usuario** | **Descripción** |
| **Nivel de privilegios** | * Poder visualizar reportes con informacion personalizada por el super usuario. |
| **Roles** | * Directivo: Tiene la disponibilidad de evidenciar el progreso de los proyectos mediante los reportes personalizados. |
| **Nivel de experiencia técnica** | Se necesita una experiencia muy básica ya que solo estará mirando el progreso de los proyectos mediante un apartado gestionado por el super usuario. |
| **Frecuencia de uso** | Cada vez que necesita visualizar el avance de los proyectos. |

Tabla 22. Caracteristicas Directivo

**Administrador**

|  |  |
| --- | --- |
| **Característica del usuario** | **Descripción** |
| **Nivel de privilegios** | * Poder asignar variables de recolección de informacion a los proyectos. * Poder asignar tipos de formularios o informes para su uso en los proyectos. * Poder crear tipos de formularios o informes. * Poder asignar coordinador y profesionales campo a cada proyecto. * Poder asignar módulos (Modulo de carga, módulo de validación, etc.) a los proyectos. |
| **Roles** | * Administrador: Capaz de gestionar la parte especifica de cada proyecto. |
| **Nivel de experiencia técnica** | Se necesita una experiencia media para el uso de los diferentes apartados que va a tener al crear un proyecto. |
| **Frecuencia de uso** | Cada vez que necesite crear un proyecto. |

Tabla 23. Caracteristicas Administrador

**Super Usuario**

|  |  |
| --- | --- |
| **Característica del usuario** | **Descripción** |
| **Nivel de privilegios** | * Poder asignar roles a los usuarios. * Tener acceso a todos los datos existentes de la plataforma. |
| **Roles** | * Administrador: Capaz de gestionar la parte especifica de cada proyecto. * Super usuario: Capaz de hacer lo mismo que el administrador y además gestionar la informacion y los usuarios que estan dentro de la plataforma. |
| **Nivel de experiencia técnica** | Se necesita una experiencia media ya que tendrá acceso a varios módulos dentro de la plataforma. |
| **Frecuencia de uso** | Cada vez que necesite darle permisos a un usuario o cuando se requiera crear un proyecto y gestionarlo. |

Tabla 24. Caracteristicas Super usuario

**7.3 Requisitos no funcionales.**

**Restricciones Generales:**

• Debe estar desarrollada en el idioma español.

• Debe tener una interfaz fácil de usar, sin muchos botones ni ventanas.

• Debe tener una interfaz agradable a la vista.

• Implementar sistemas de seguridad.

• El sitio debe evitar el acceso a páginas que requieran una autenticación previa.

**Restricciones de Software:**

• Se debe realizar haciendo uso del framework Angular.

• Se debe hacer uso de HMTL para el front-end.

• Se debe hacer uso de CSS para el front-end.

• Se debe hacer uso de Typescript para el front-end.

• Se debe hacer uso de JavaScript para el back-end.

• Debe poder abrirse en cualquier navegador web.

**Restricciones de Hardware:**

• La aplicación debe poder ejecutarse tanto en computadores de escritorio como en computadores portátiles.

• Sistema operativo mínimo: Windows xp el cual tenga acceso a internet.

• Con una velocidad estable mínimo de descarga de 2mb/s puede ejecutarse la aplicación web.

• El computador debe tener 1gb de memoria RAM como mínimo.

• La resolución de la pantalla puede ser cualquiera.

1. **Plan de trabajo.**

**8.1 Mapa de historias de usuario.**

En esta sección el equipo encargado de implementar la plataforma deberá realizar el mapa de historias de usuario, en el cual, dependiendo al equipo, el desarrollo de las historias de usuario se dividirá en backlogs. Aparte de realizar el mapa de historias de usuario, se recomienda usar alguna herramienta para la gestion de los backlogs y springs como trello o alguna similar para llevar un correcto orden del proceso de desarrollo.

**8.2 Métodos y herramientas de estimación.**

Se debe realizar la estimación de desarrollo de las diferentes historias de usuario, con el fin de estimar el tiempo de desarrollo de todo el proyecto, el uso de herramientas como por ejemplo planning póker sirve para desarrollar la siguiente tabla.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tareas (Historias de usuario)** | **Esfuerzo por persona en horas.** | **Duración en días** | **Dependencias.** |
| Tarea 1 - Transversalidad en los datos |  |  |  |
| Tarea 2 - Espacio centralizado donde se va a encontrar toda la información. |  |  |  |
| Tarea 3 - Etiquetado de la información. |  |  |  |
| Tarea 4 - Historial de las personas. |  |  |  |
| Tarea 5 - Descargar reportes de la plataforma. |  |  |  |
| Tarea 6 - Subida de informacion a la plataforma. |  |  |  |
| Tarea 7 - Validación y ayuda de llenado de información. |  |  |  |
| Tarea 8 - Verificación de subida de la información a la plataforma. |  |  |  |
| Tarea 9 - Indicadores operativos. |  |  |  |
| Tarea 10 - Retroalimentación o solicitudes de corrección. |  |  |  |
| Tarea 11 - Verificar el cumplimiento de los profesionales de campo. |  |  |  |
| Tarea 12 - Apartado de tareas pendientes. |  |  |  |
| Tarea 13 - Gestion de permisos. |  |  |  |
| Tarea 14 - Gestion especifica de cada proyecto. |  |  |  |
| Tarea 15 - Reportes para directivos. |  |  |  |
| Tarea 16 - Acceso a toda la información. |  |  |  |
| Tarea 17 - Trazabilidad de herramientas utilizadas. |  |  |  |

Tabla 25. Métodos y herramientas de estimación

1. **Arquitectura.**
   1. **Vista lógica del sistema.**
      1. **Diagrama de componentes**

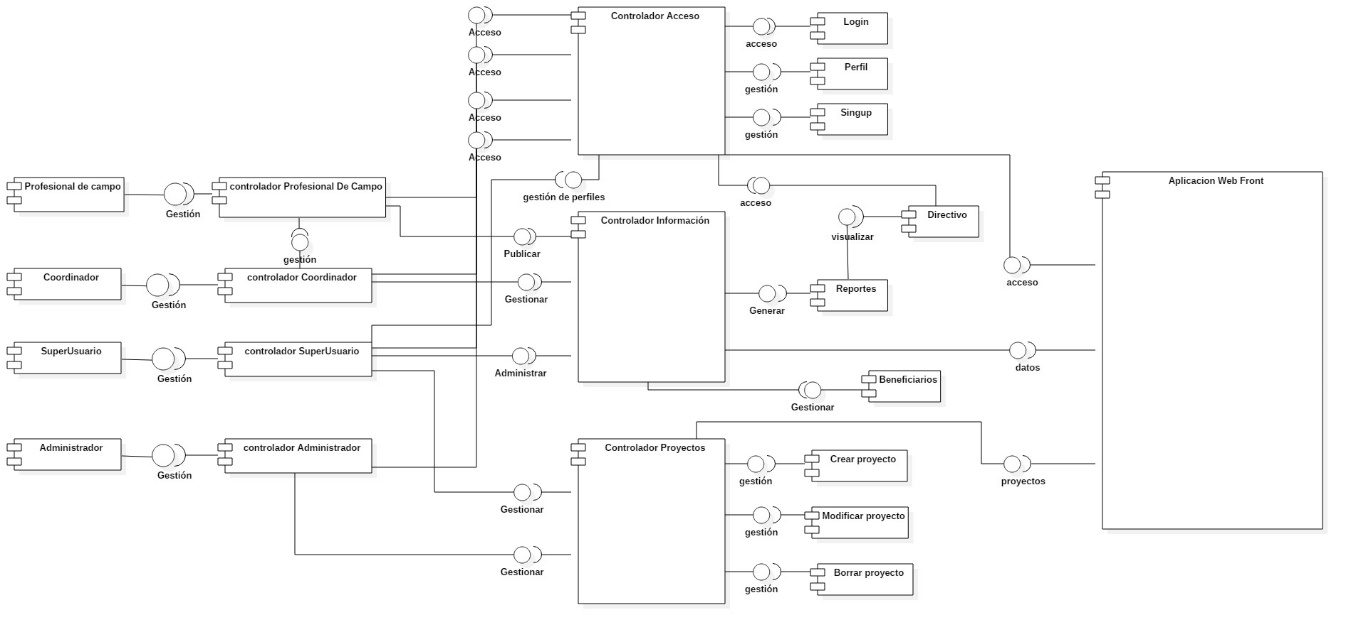
****

Figura 4. Diagrama de componentes

En este diagrama (figura 4) se muestra el diagrama de componentes, en donde vemos la presencia de todos los usuarios representados individualmente, esto es debido a que cada uno hace tareas distintas, por lo cual es recomendable tener un controlador para cada uno de estos, a excepción del usuario directivo el cual no tiene un controlador ya que su única funcion es visualizar los reportes. Otro punto para destacar es que tenemos tres controladores principales, el de acceso es el cual autentica y les da acceso a los usuarios, este puede ser modificado por el super usuario, por ejemplo, el super usuario puede asignarles permisos a los diferentes usuarios. Por otro lado tenemos al controlador información, este posee toda la informacion del sistema, y es donde se habilitan las funcionalidades de subir informacion, gestionar la información y administrarla, además en este controlador se debe poder generar reportes para los directivos. Y por último tenemos al controlador proyectos en donde el administrador y el super usuario tienen la disponibilidad de crear, modificar y eliminar parámetros de los proyectos.

* + 1. **Diagrama de paquetes.**

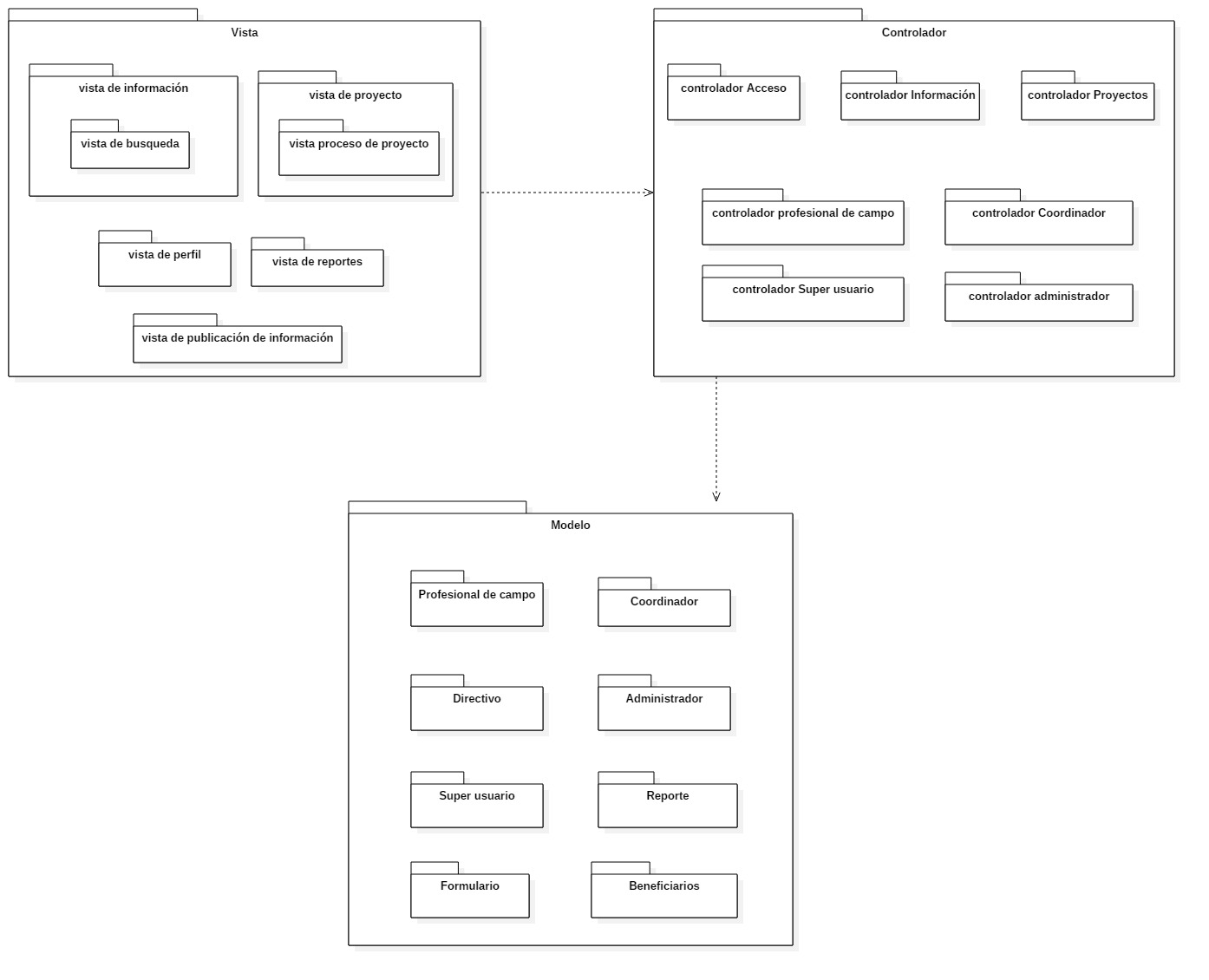
****

Figura 5. Diagrama de paquetes

**Paquete vista:** En este paquete encontramos las clases relacionadas a la interfaz gráfica, donde vemos la vista del proyecto en general y además el proceso de este proyecto, donde se incluyen las tareas pendientes del proyecto, información de este, usuarios que estan participando, etc.

**Paquete controlador:** En este paquete esta básicamente toda la lógica del negocio, es decir, donde estan las funcionalidades de cada tipo de usuario. Tambien esta presente el manejo de toda la información del negocio, en donde se nutre y se gestiona por los usuarios presentes en la lógica del negocio.

**Paquete modelo:** En este paquete estan presentes las entidades de los diferentes usuarios, y la representación de reportes y formularios que hacen parte de del negocio.

* 1. **Vista física del sistema.**
     1. **Diagrama de despliegue.**

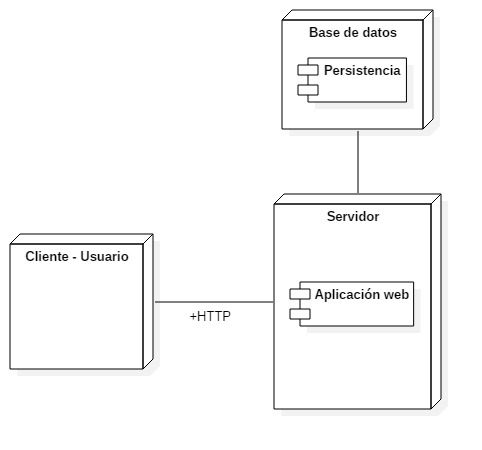
****

Figura 6. Diagrama de despliegue

**Nodo base de datos:** Este nodo básicamente representa la base de datos que se va a utilizar para el funcionamiento de la aplicación, es por ello por lo que se conecta al servidor.

**Nodo servidor:** Este nodo representa el servidor que va a contener la aplicación web y que va a ser accedo por lo usuarios mediante una comunicación http.

**Nodo cliente – usuario:** Este nodo representa al dispositivo que se va a usar para acceder a la aplicación web, el dispositivo puede ser un celular y/o un pc los cuales tengan acceso a internet y cuenten con un navegador para acceder a la aplicación.

* 1. **Diseño detallado.**
     1. **Diagrama de clases**

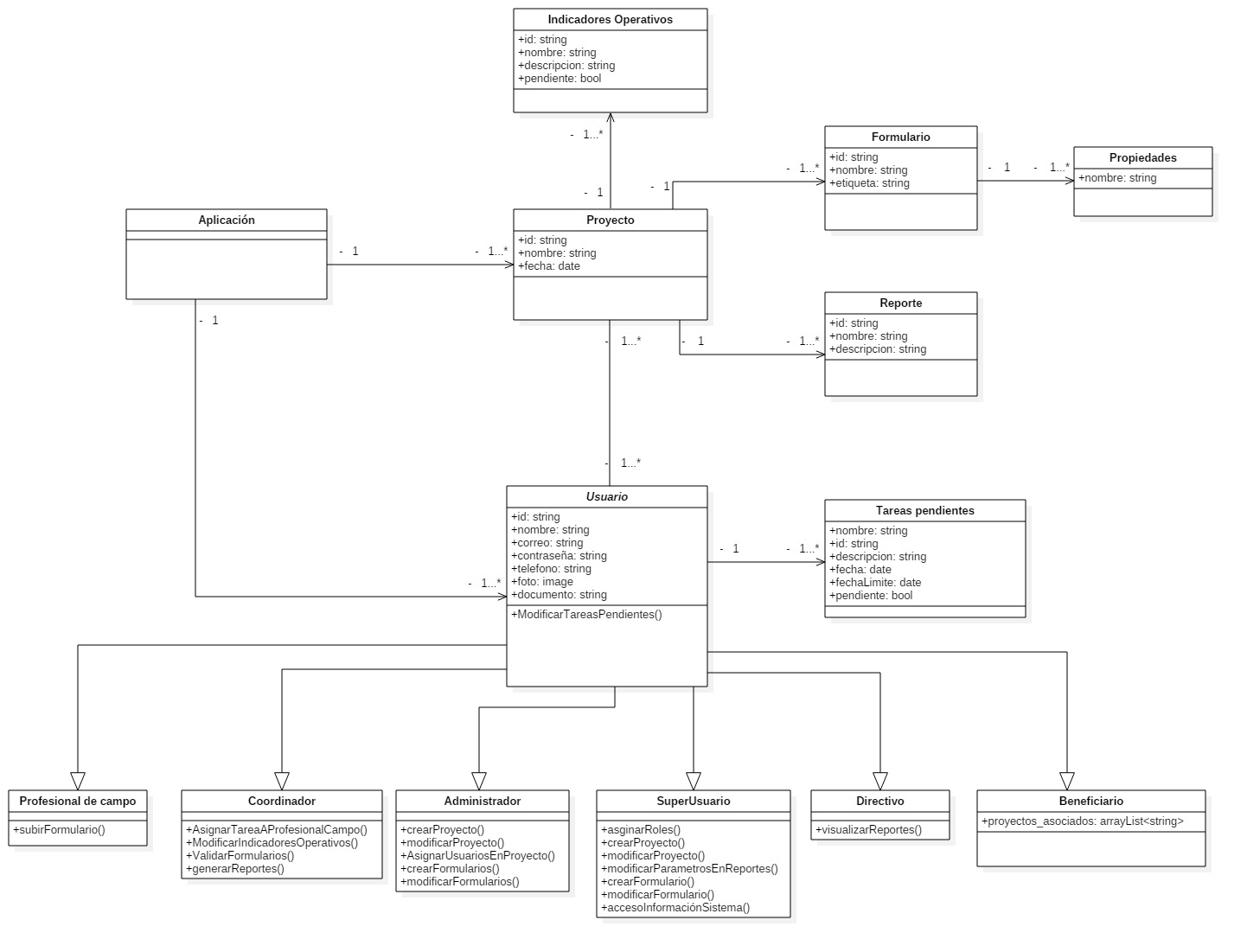
****

Figura 7. Diagrama de clases

En este diagrama de clases podemos ver la aplicación que tiene 1 o varios proyectos, el cual posee su nombre, su id y su fecha, además cada proyecto tiene 1 o varios indicadores operativos el cual tiene un parámetro booleano el cual es pendiente y nos indica si esta cumplido o no, tambien cada proyecto tiene 1 o varios formularios y su vez cada formulario tiene una lista de propiedades los cuales nos sirven para poder modificar un formulario, y por ultimo cada proyecto tiene 1 o varios reportes que se generan en base a la información que este dentro de cada proyecto. Además dentro de la aplicación tenemos 1 o varios usuarios que pueden ser diferentes tipos de usuarios, tambien cada tipo de usuario tiene diferentes métodos ya que cada uno cumple una función diferente dentro de la aplicación. Otro punto importante para destacar es que tenemos una relación unidireccional entre usuario y proyecto, la cual nos indica que un proyecto puede tener varios usuarios y un usuario puede estar en varios proyectos, esta relación tambien nos sirve para poder implementar los métodos que puede realizar el administrador y super usuario de crear proyectos, o modificar formularios, etc.

* + 1. **Persistencia.**

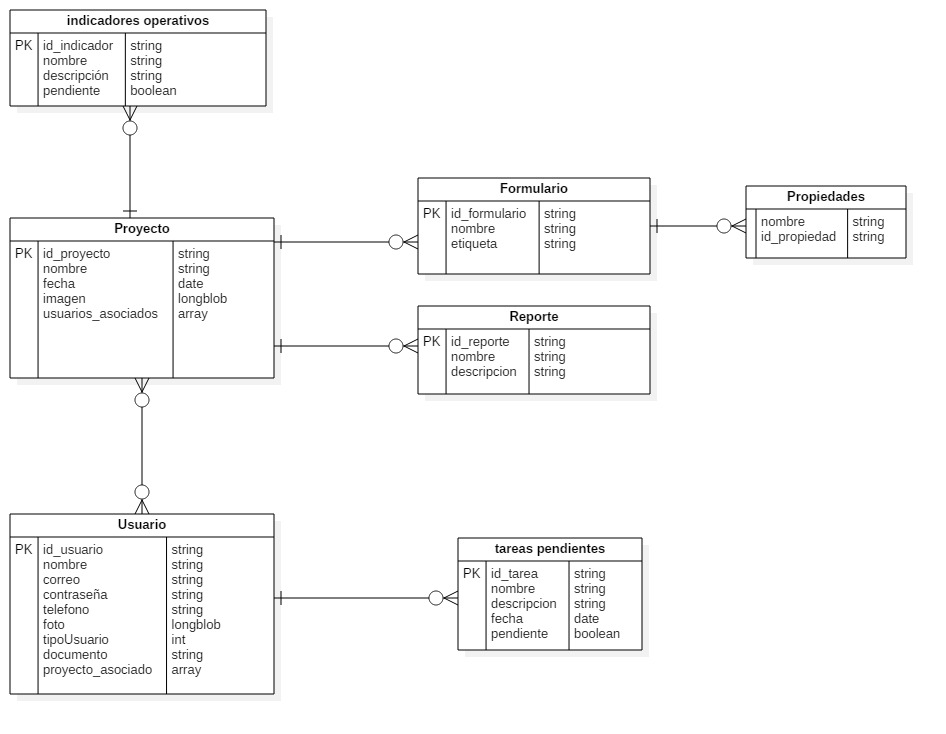
****

Figura 8. Diagrama de persistencia

En la figura 8 se puede evidenciar como será la persistencia de los datos, en donde tenemos una relacion muchos a muchos entre proyecto y usuario. Tambien cada proyecto tiene relacion uno a mucho con los diferentes formularios que existan en este, y a su vez cada formulario tiene una relacion 1 a muchos con propiedades, que básicamente la idea es que se tenga un banco de propiedades para que podamos personalizar un formulario. Tambien proyecto tendrá una relación uno a muchos con indicadores operativos, que básicamente sirve como una plan de acción para ejecutar el proyecto. Usuario tambien tiene una relacion 1 a muchos a tareas pendientes que representa la lista de tareas pendiente que tendrá cada usuario. La representación de los diferentes usuarios se hará por medio de un int tipo de usuario ya que todos los usuarios tendrán la misma información asociada.

**9.4 Interfaz de usuario.**

1. **Anexos.**

En esta sección se encontrarán los recursos que forman parte del diseño.

* Miro Historias de usuario.

<https://miro.com/welcomeonboard/ejU1UWNlcjFYMmFEVDVIcm9VUGlzSFhKazlRQ1libVVIMXlnT240SVR6S0lWa0pSaUVvdFl6Nk1HckQ0cFFoWHwzMDc0NDU3MzU0Mjg1NzM5Nzky?invite_link_id=871645957622>

* Repositorio GitHub.

https://github.com/Fabian-Olarte/JCF\_Proyecto