SOFTWARE DE FLUJO DE TRABAJO VIRTUAL PARA LA UNIVERSIDAD DE MANIZALES

ANDRIUS CAMILO SOTO GRISALES

UNIVERSIDAD DE MANIZALES

FACULTAD DE CIENCIAS E INGENIERIA

INGENIERIA DE SISTEMAS Y TELECOMUNICACIONES

MAINZALES

2020

SOFTWARE DE FLUJO DE TRABAJO VIRTUAL PARA LA UNIVERSIDAD DE MANIZALES

ANDRIUS CAMILO SOTO GRISALES

MODELO PRESENTACION DE PROPUESTA DE PRACTICA

JORGE HERNAN FRANCO FRANCO

UNIVERSIDAD DE MANIZALES

FACULTAD DE CIENCIAS E INGENIERIA

INGENIERIA DE SISTEMAS Y TELECOMUNICACIONES

MANIZALES

2020

CONTENIDO

Pág.

Contenido

Introducción………………………………………………………………………….…………...…1

[1. Justificación](#_Toc32906547) 2

[2. Area problematica](#_Toc32906548) 3

[2.1. Definición del problema](#_Toc32906549) 3

[3. Objetivos](#_Toc32906544) 4

[3.1. Objetivo general](#_Toc32906545) 4

[3.2. Objetivos especificos](#_Toc32906546) 4

[4. Metodologia](#_Toc32906550) 5

[4.1. Procedimiento](#_Toc32906551) 5

[5. Cronograma](#_Toc32906555) 7

[6. Marco teorico](#_Toc32906556) 8

[7. Desarrollo de proyecto](#_Toc32906558) 9

8. Webgrafia……………………………………………………………………………………….10

INTRODUCCIÓN

La Universidad de Manizales presenta la siguiente reseña a cerca de su labor:

“*Es una institución educativa que centra sus recursos y compromisos al ámbito educativo profesional teniendo siempre presente la finalidad de formar profesionales, aportando opciones académicas tanto de pregrado como posgrado y educación continuada”.[[1]](#footnote-1)*

El flujo de datos que genera la empresa es de gran magnitud, pues es una empresa que vincula personal y estudiantes nuevos cada determinado periodo de tiempo en grandes masas al menos en lo que a estudiantes se refiere, además, los registros ya puestos sobre las bases de datos que maneja la institución son expuestos a sufrir modificaciones cada que sea necesario, por todo lo anterior es que la empresa se prepara para la adecuada gestión de dichos datos que se registran diariamente, para ello es necesario contar tanto con personal como con infraestructura adecuada. La empresa cuenta con su debido espacio dentro de algunas dependencias para manejar datos específicos, así como también cuenta con centro de datos equipado con infraestructura pertinente como servidores encargados de alimentar y permitir el flujo de datos en el interior de la empresa, así mismo existe dentro de la empresa sistemas informáticos y aplicaciones que permiten la gestión de los datos.

La empresa actualmente cuenta con un sistema virtual con la capacidad de permitir realizar varias gestiones institucionales a los usuarios tanto estudiantes de la institución como empleados de la misma, dicho sistema permite al empleado varias acciones dentro de la plataforma web en donde se desataca uno de los aplicativos que tiene la plataforma virtual dedicado al flujo de solicitudes y/o tramites de los empleados de manera laboral. El aplicativo tiene como funcionalidades principales la gestión y creación de solicitudes o trámites internos de trabajo

## JUSTIFICACIÓN

El proyecto de flujo trabajo virtual debe de hacerse en la empresa porque la plataforma virtual con la que cuentan en la institución no satisface por completo las necesidades de los usuarios, específicamente hablando del aplicativo de flujo de trabajo que se maneja actualmente, lo anterior, debido a que primero el aplicativo no posibilita la opción de recolectar registros y reportes estadísticos respecto al flujo de solicitudes que se lleva a cabo dentro del sistema virtual, también la parte visual del aplicativo carece de brindar la posibilidad de una completa comprensión de funcionamiento para el usuario, por otro lado está la automatización del aplicativo actualmente carece de fluidez en cuanto a la carga de datos que se registra dentro del aplicativo, lo cual genera retrasos en las respuestas del aplicativo.

# 2. AREA PROBLEMATICA

## 2.1. DEFINICIÓN DEL PROBLEMA

La empresa actualmente cuenta con un sistema virtual con la capacidad de permitir realizar varias gestiones institucionales a los usuarios tanto estudiantes de la institución como empleados de la misma, dicho sistema permite al empleado varias acciones dentro de la plataforma web en donde se desataca uno de los aplicativos que tiene la plataforma virtual dedicado al flujo de solicitudes y/o tramites de los empleados de manera laboral. El aplicativo tiene como funcionalidades principales la gestión y creación de solicitudes o trámites internos de trabajo.

El aplicativo actual tiene problemas como:

* No estar del todo automatizado, pues el flujo de las solicitudes es direccionado manualmente.
* No contar con un nivel adecuado para brindar visualmente comprensión de funcionalidad al usuario.
* Contar con tecnologías de desarrollo con un nivel alto de consumo de recursos.
* Tener formatos de redacción de solicitudes no dinámicas, lo cual no permite considerar la opción de generar registros estadísticos acerca del uso del aplicativo.

# 3. OBJETIVOS

## 3.1. Objetivo general

Diseñar un aplicativo web que permita el flujo de solicitudes internas de trabajo dentro de la plataforma web de la Universidad de Manizales (Sigum) de manera que se pueda actualizar y mejorar el aplicativo actual que maneja la plataforma web de la institución.

## 3.2. Objetivos específicos

* + 1. Definir los módulos y desarrollar el debido análisis del aplicativo a desarrollar.
    2. Desarrollar el diseño que tendrá el aplicativo.
    3. Definir y desarrollar la organización estructural y los tipos de solicitud.
    4. Definir y diseñar el flujo respectivo que tendrá cada tipo de solicitud dentro del aplicativo.
    5. Definir y diseñar un proceso que permita gestionar los subprocesos personales tales como crear solicitud, visualizar solicitudes pendientes de aprobación y ver notificaciones.

1. METODOLOGÍA

Sinnaps presenta la siguiente definición a cerca de metodología Scrum:

“*Scrum es un método para trabajar en equipo a partir de iteraciones.*

*Así pues, Scrum es una metodología ágil, por lo que su objetivo será controlar y planificar proyectos con un gran volumen de cambios de última hora, en donde la incertidumbre sea elevada.*

*Se suele planificar por semanas. Al final de cada Sprint o iteración, se va revisando el trabajo validado de la anterior semana. En función de esto, se priorizan y planifican las actividades en las que invertiremos nuestros recursos en el siguiente Sprint”.[[2]](#footnote-2)*

4.1. Procedimiento

Sprint 1.

Plan de proyectos. Se lleva a cabo una serie de procesos que ayudan a clarificar el alcance del proyecto y de esta forma planificar de manera general el contenido del aplicativo.

Sprint 2.

Modulo 1: Estructura organizacional.

Fase 1: Historias de usuario.

1. Definir requerimientos. Se desarrolla una descripción completa del comportamiento del aplicativo que se va a desarrollar.
2. Definir tipos de usuario. Se lleva a cabo una descripción del comportamiento del aplicativo desde el punto de vista de cada tipo de usuario que manipulará el sistema.

Fase 2: Planeación.

1. Plan proyecto. Se desarrolla la planificación y organización de los elementos que contiene el proyecto dentro del primer módulo.
2. Análisis.
   1. Casos de uso. Se lleva a cabo el desarrollo de la especificación y comportamiento del proyecto dentro del primer módulo mediante su interacción con los usuarios o actores independientes del aplicativo.
3. Diseño.
   1. Modelo entidad relación. Se desarrolla la descripción de los datos contenidos en el aplicativo, descripción tanto del tipo de dato como de las relaciones que se tienen entre dichos datos y/o usuarios.
   2. Diagrama de actividades. Se desarrolla el modelado del comportamiento que tendrá el aplicativo enfatizando en el nivel jerárquico de los procesos de funcionabilidad.

Fase 3. Desarrollo.

1. Back-end. Se desarrolla la codificación necesaria en la fase del código que se encarga de la lógica y conexión a base de datos.
2. Front-end. Se desarrollan los elementos necesarios dentro del aplicativo que permiten la interacción del usuario con el mismo.
3. Pruebas. Se llevan a cabo una serie de procesos que ponen a prueba cada funcionalidad y entorno de seguridad con la que cuenta el aplicativo.

Sprint 3.

Modulo 2: Tipos de solicitud.

Fase 1: Historias de usuario.

1. Definir requerimientos. Se desarrolla una descripción completa del comportamiento del aplicativo que se va a desarrollar.
2. Definir tipos de usuario. Se lleva a cabo una descripción del comportamiento del aplicativo desde el punto de vista de cada tipo de usuario que manipulará el sistema.

Fase 2: Planeación.

1. Plan proyecto. Se desarrolla la planificación y organización de los elementos que contiene el proyecto dentro del primer módulo.
2. Análisis.
   1. Casos de uso. Se lleva a cabo el desarrollo de la especificación y comportamiento del proyecto dentro del primer módulo mediante su interacción con los usuarios o actores independientes del aplicativo.
3. Diseño.
   1. Modelo entidad relación. Se desarrolla la descripción de los datos contenidos en el aplicativo, descripción tanto del tipo de dato como de las relaciones que se tienen entre dichos datos y/o usuarios.
   2. Diagrama de actividades. Se desarrolla el modelado del comportamiento que tendrá el aplicativo enfatizando en el nivel jerárquico de los procesos de funcionabilidad.

Fase 3. Desarrollo.

1. Back-end. Se desarrolla la codificación necesaria en la fase del código que se encarga de la lógica y conexión a base de datos.
2. Front-end. Se desarrollan los elementos necesarios dentro del aplicativo que permiten la interacción del usuario con el mismo.
3. Pruebas. Se llevan a cabo una serie de procesos que ponen a prueba cada funcionalidad y entorno de seguridad con la que cuenta el aplicativo.

Sprint 4.

Modulo 3: Flujo de solicitudes.

Fase 1: Historias de usuario.

1. Definir requerimientos. Se desarrolla una descripción completa del comportamiento del aplicativo que se va a desarrollar.
2. Definir tipos de usuario. Se lleva a cabo una descripción del comportamiento del aplicativo desde el punto de vista de cada tipo de usuario que manipulará el sistema.

Fase 2: Planeación.

1. Plan proyecto. Se desarrolla la planificación y organización de los elementos que contiene el proyecto dentro del primer módulo.
2. Análisis.
   1. Casos de uso. Se lleva a cabo el desarrollo de la especificación y comportamiento del proyecto dentro del primer módulo mediante su interacción con los usuarios o actores independientes del aplicativo.
3. Diseño.
   1. Modelo entidad relación. Se desarrolla la descripción de los datos contenidos en el aplicativo, descripción tanto del tipo de dato como de las relaciones que se tienen entre dichos datos y/o usuarios.
   2. Diagrama de actividades. Se desarrolla el modelado del comportamiento que tendrá el aplicativo enfatizando en el nivel jerárquico de los procesos de funcionabilidad.

Fase 3. Desarrollo.

1. Back-end. Se desarrolla la codificación necesaria en la fase del código que se encarga de la lógica y conexión a base de datos.
2. Front-end. Se desarrollan los elementos necesarios dentro del aplicativo que permiten la interacción del usuario con el mismo.
3. Pruebas. Se llevan a cabo una serie de procesos que ponen a prueba cada funcionalidad y entorno de seguridad con la que cuenta el aplicativo.

Sprint 5.

Modulo 4: Gestión de solicitudes

Fase 1: Historias de usuario.

1. Definir requerimientos. Se desarrolla una descripción completa del comportamiento del aplicativo que se va a desarrollar.
2. Definir tipos de usuario. Se lleva a cabo una descripción del comportamiento del aplicativo desde el punto de vista de cada tipo de usuario que manipulará el sistema.

Fase 2: Planeación.

1. Plan proyecto. Se desarrolla la planificación y organización de los elementos que contiene el proyecto dentro del primer módulo.
2. Análisis.
   1. Casos de uso. Se lleva a cabo el desarrollo de la especificación y comportamiento del proyecto dentro del primer módulo mediante su interacción con los usuarios o actores independientes del aplicativo.
3. Diseño.
   1. Modelo entidad relación. Se desarrolla la descripción de los datos contenidos en el aplicativo, descripción tanto del tipo de dato como de las relaciones que se tienen entre dichos datos y/o usuarios.
   2. Diagrama de actividades. Se desarrolla el modelado del comportamiento que tendrá el aplicativo enfatizando en el nivel jerárquico de los procesos de funcionabilidad.

Fase 3. Desarrollo.

1. Back-end. Se desarrolla la codificación necesaria en la fase del código que se encarga de la lógica y conexión a base de datos.
2. Front-end. Se desarrollan los elementos necesarios dentro del aplicativo que permiten la interacción del usuario con el mismo.
3. Pruebas. Se llevan a cabo una serie de procesos que ponen a prueba cada funcionalidad y entorno de seguridad con la que cuenta el aplicativo.

## CRONOGRAMA.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | *Actividad* | Febrero | | | | Marzo | | | | | Abril | | | | Mayo | | | | Junio | | | | Julio | | | |
| S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | | S10 | S11 | S12 | S13 | S14 | S15 | S16 | S17 | S18 | S19 | S20 | S21 | S22 | S23 | S24 |
| 1 | Sprint 1 | **x** | **x** |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.2 | Plan de proyecto | **x** | **x** |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Sprint 2 |  | **x** | **x** | **x** | **x** | **x** | **x** | **x** | **x** | | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1 | Modulo 1: Estructura organizacional |  | **x** | **x** | **x** | **x** | **x** | **x** | **x** | **x** | | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1.1 | Fase 1. Historias de usuario |  | **x** | **x** |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1.1.1 | Definir requerimientos |  | **x** |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1.1.2 | Definir tipos de usuario |  |  | **x** |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1.2 | Fase 2. Planeación |  |  | **x** | **x** |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1.2.1 | Plan proyecto del modulo |  |  | **x** |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1.2.2 | Análisis |  |  | **x** | **x** |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1.2.2.1 | Casos de uso |  |  | **x** | **x** |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1.2.3 | Diseño |  |  |  | **x** |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1.2.3.1 | Diagrama de actividades |  |  |  | **x** |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1.2.3.2 | Diccionario de datos |  |  |  | **x** |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1.2.3.3 | Modelo entidad-relación |  |  |  | **x** |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1.3 | Fase 3. Desarrollo |  |  |  |  | **x** | **x** | **x** | **x** | **x** | | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1.3.1 | Back-end |  |  |  |  | **x** | **x** |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1.3.2 | Front-end |  |  |  |  |  |  | **x** | **x** | **x** | | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1.3.3 | Pruebas |  |  |  |  |  |  |  |  |  | | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Sprint 3 |  |  |  |  |  |  |  |  |  | | **x** | **x** | **x** |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1 | Modulo 2: Tipos de solicitud |  |  |  |  |  |  |  |  |  | | **x** | **x** | **x** |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1.1 | Fase 1. Historias de usuario |  |  |  |  |  |  |  |  |  | | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1.1.1 | Definir requerimientos |  |  |  |  |  |  |  |  |  | | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1.1.2 | Definir tipos de usuario |  |  |  |  |  |  |  |  |  | | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1.2 | Fase 2. Planeación |  |  |  |  |  |  |  |  |  | | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1.2.1 | Plan proyecto del modulo |  |  |  |  |  |  |  |  |  | | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1.2.2 | Análisis |  |  |  |  |  |  |  |  |  | | **x** | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1.2.2.1 | Casos de uso |  |  |  |  |  |  |  |  |  | |  | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1.2.3 | Diseño |  |  |  |  |  |  |  |  |  | |  | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1.2.3.1 | Modelo entidad-relación |  |  |  |  |  |  |  |  |  | |  | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1.2.3.2 | Diccionario de datos |  |  |  |  |  |  |  |  |  | |  | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1.2.3.3 | Diagrama de actividades |  |  |  |  |  |  |  |  |  | |  | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1.3 | Fase 3. Desarrollo |  |  |  |  |  |  |  |  |  | |  | **x** | **x** |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1.3.1 | Back-end |  |  |  |  |  |  |  |  |  | |  |  | **x** |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1.3.2 | Front-end |  |  |  |  |  |  |  |  |  | |  |  | **x** |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1.3.3 | Pruebas |  |  |  |  |  |  |  |  |  | |  |  | **x** |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Sprint 4 |  |  |  |  |  |  |  |  |  | |  |  |  | **x** | **x** | **x** | **x** | **x** | **x** | **x** |  |  |  |  |  |
| 4.1 | Modulo 3: Flujo de solicitudes |  |  |  |  |  |  |  |  |  | |  |  |  | **x** | **x** | **x** | **x** | **x** | **x** | **x** |  |  |  |  |  |
| 4.1.1 | Fase 1. Historias de usuario |  |  |  |  |  |  |  |  |  | |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  |  |
| 4.1.1.1 | Definir requerimientos |  |  |  |  |  |  |  |  |  | |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  |  |
| 4.1.1.2 | Definir tipos de usuario |  |  |  |  |  |  |  |  |  | |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  |  |
| 4.1.2 | Fase 2. Planeación |  |  |  |  |  |  |  |  |  | |  |  |  | **x** | **x** |  |  |  |  |  |  |  |  |  |  |
| 4.1.2.1 | Plan proyecto del modulo |  |  |  |  |  |  |  |  |  | |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  |  |
| 4.1.2.2 | Análisis |  |  |  |  |  |  |  |  |  | |  |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  |
| 4.1.2.2.1 | Casos de uso |  |  |  |  |  |  |  |  |  | |  |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  |
| 4.1.2.3 | Diseño |  |  |  |  |  |  |  |  |  | |  |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  |
| 4.1.2.3.1 | Modelo entidad-relación |  |  |  |  |  |  |  |  |  | |  |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  |
| 4.1.2.3.2 | Diccionario de datos |  |  |  |  |  |  |  |  |  | |  |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  |
| 4.1.2.3.3 | Diagrama de actividades |  |  |  |  |  |  |  |  |  | |  |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  |
| 4.1.3 | Fase 3. Desarrollo |  |  |  |  |  |  |  |  |  | |  |  |  |  | **x** | **x** | **x** | **x** | **x** | **x** |  |  |  |  |  |
| 4.1.3.1 | Back-end |  |  |  |  |  |  |  |  |  | |  |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  |
| 4.1.3.2 | Front-end |  |  |  |  |  |  |  |  |  | |  |  |  |  |  | **x** | **x** | **x** | **x** | **x** |  |  |  |  |  |
| 4.1.3.3 | Pruebas |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  | **x** |  |  |  |  |  |
| 5 | Sprint 5 |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  | **x** | **x** | **x** | **x** | **x** |  |
| 5.1 | Modulo 4: Gestión de solicitudes |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  | **x** | **x** | **x** | **x** | **x** |  |
| 5.1.1 | Fase 1. Historias de usuario |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  | **x** |  |  |  |  |  |
| 5.1.1.1 | Definir requerimientos |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  | **x** |  |  |  |  |  |
| 5.1.1.2 | Definir tipos de usuario |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  | **x** |  |  |  |  |  |
| 5.1.2 | Fase 2. Planeación |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  | **x** |  |  |  |  |  |
| 5.1.2.1 | Plan proyecto del modulo |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  | **x** |  |  |  |  |  |
| 5.1.2.2 | Análisis |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  | **x** | **x** |  |  |  |  |
| 5.1.2.2.1 | Casos de uso |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  | **x** | **x** |  |  |  |  |
| 5.1.2.3 | Diseño |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  | **x** |  |  |  |  |
| 5.1.2.3.1 | Modelo entidad-relación |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  | **x** |  |  |  |  |
| 5.1.2.3.2 | Diagrama de actividades |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  | **x** |  |  |  |  |
| 5.1.3 | Fase 3. Desarrollo |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  | **x** | **x** | **x** | **x** |  |
| 5.1.3.1 | Back-end |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  | **x** | **x** |  |  |  |
| 5.1.3.2 | Front-end |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  | **x** | **x** |  |
| 5.1.3.3 | Pruebas |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  | **x** |  |

1. MARCO TEORICO.

Como antecedente se toma el siguiente trabajo realizado por Pinto, C. A.:

“*Eleval decidió llevar adelante un proyecto piloto para automatizar los procesos de solicitudes utilizando cinco formularios de la unidad de organización y recursos humanos. Eleval requiere que esta solución sea desarrollada usando Microsoft Office InfoPath 2007 y que este enmarcada dentro de la internet corporativa bajo las tecnologías de Microsoft Office SharePoint Server. El proyecto piloto permitirá comprender las implicaciones de un proyecto de este tipo, evaluar si se satisfacen todos los requerimientos de la organización y mitigar los problemas que pudieran encontrarse durante la implementación, antes de llevar adelante el proyecto para toda la empresa”.*[[3]](#footnote-3)

El trabajo anteriormente citado se acerca mucho al propósito que tiene el aplicativo de flujo de trabajo, que tiene como objetivo entre otras cosas automatizar los flujos de trabajo que se crean dentro del sistema con los usuarios de la empresa. En cuanto a metodología se diferencian mucho tanto el trabajo citado como el aplicativo que se quiere llevar a cabo, pues son tanto procesos como herramientas diferentes para combatir ambos objetivos generales.

6.1. Metodología SCRUM.

Sinnaps presenta la siguiente definición a cerca de metodología Scrum:

“*Scrum es un método para trabajar en equipo a partir de iteraciones.*

*Así pues, Scrum es una metodología ágil, por lo que su objetivo será controlar y planificar proyectos con un gran volumen de cambios de última hora, en donde la incertidumbre sea elevada.*

*Se suele planificar por semanas. Al final de cada Sprint o iteración, se va revisando el trabajo validado de la anterior semana. En función de esto, se priorizan y planifican las actividades en las que invertiremos nuestros recursos en el siguiente Sprint”.*[[4]](#footnote-4)

Se suele planificar por semanas. Al final de cada Sprint o iteración, se va revisando el trabajo validado de la anterior semana. En función de esto, se priorizan y planifican las actividades en las que invertiremos nuestros recursos en el siguiente Sprint

* 1. Node.js.

[[5]](#footnote-5)Es un entorno de ejecución para “JavaScript”[[6]](#footnote-6) construido con el motor de JavaScript V8 de “Chrome”[[7]](#footnote-7).

Node.js tiene un diseño similar y está influenciado por sistemas como “Twisted de Python”[[8]](#footnote-8). Node.js lleva el modelo de eventos un poco más allá, este presenta un [bucle de eventos](https://github.com/nodejs/node/blob/master/doc/topics/event-loop-timers-and-nexttick.md) como un entorno en vez de una librería. En otros sistemas siempre existe una llamada que bloquea para iniciar el bucle de eventos.

* 1. Bootstrap.

[[9]](#footnote-9)Bootstrap es un kit de herramientas de código abierto para desarrollar con HTML, CSS y JS. Realice rápidamente prototipos de sus ideas o cree toda su aplicación con nuestras variables, sistema de cuadrícula sensible, componentes precompilados extensos y complementos potentes creados en “jQuery”[[10]](#footnote-10).

1. DESARROLLO DEL PROYECTO.

Frente a la estipulación de fechas en el cronograma de cumplimiento se presentan a continuación los procesos que han culminado su desarrollo en el progreso actual del mismo.

**Sprint 1.**

Plan de proyectos**.**

1. INTERESADOS/AFECTADOS

|  |  |  |
| --- | --- | --- |
| STAKEHOLDER | ROL | INFLUENCIA |
| Dependencias | Usuario primario del aplicativo | Alta |
| Empleados | Usuario primario del aplicativo | Alta |
| Administrador | Usuario primario del aplicativo | Alta |

# RECURSOS ASIGNADOS DE FORMA PREVIA

* 1. RECURSOS DE HUMANOS:

Personal del área de tecnologías de la información de la Universidad de Manizales

# REQUERIMIENTOS DE LOS INTERESADOS

* 1. REQUERIMIENTOS DEL PROYECTO
* Módulo 1: Estructura organizacional y tipos de solicitud

1. Administrador.

El aplicativo permitirá.

* 1. Dependencias.
     1. Administrar dependencias
* Crear dependencias nuevas
* Actualizar dependencias
* Eliminar dependencias
* Ingresar orden jerárquico de las dependencias
* Modificar orden jerárquico de las dependencias
  + 1. Visualizar dependencias
* Visualizar las dependencias registradas en el sistema
* Visualizar el orden jerárquico de las dependencias
* Visualizar la descripción a cerca de la dependencia seleccionada
* Visualizar en forma de árbol desplegable los cargos pertenecientes a las dependencias
  1. Cargo.
     1. Administrar los cargos.
* Crear nuevos cargos
* Asignar dependencia a cada cargo
* Actualizar cargos
* Eliminar cargos
* Ingresar orden jerárquico de los cargos
* Modificar orden jerárquico de los cargos
  + 1. Visualizar cargos
* Visualizar los cargos que se han registrado en el sistema y los usuarios que ocupan dicho cargo
* Visualizar el orden jerárquico que tienen los cargos de cada dependencia
* Visualizar la descripción a cerca de los cargos seleccionados
* Visualizar en forma de árbol desplegable las personas que ocupan dicho cargo
  1. Usuarios.
     1. Administrar usuarios
* Crear usuarios nuevos
* Asignar cargo a cada empleado
* Actualizar usuarios
* Eliminar usuarios
* Modificar orden jerárquico a los usuarios
  + 1. Visualizar usuarios
* Visualizar los usuarios registrados en el sistema
* Visualizar el orden jerárquico de los usuarios
* Visualizar la descripción a cerca de los usuarios seleccionados
  1. Tipos de solicitud
* Crear tipos de solicitud
* Actualizar tipos de solicitud
* Borrar tipos de solicitud

1. Usuario

El aplicativo permitirá.

* 1. Dependencias.
     1. Visualizar las dependencias
* Visualizar las dependencias registradas en el sistema
* Visualizar el orden jerárquico de las dependencias
* Visualizar la descripción a cerca de la dependencia seleccionada
* Visualizar en forma de árbol desplegable los cargos pertenecientes cada dependencia
  1. Cargos.
* Visualizar cargos
* Visualizar los cargos registrados en el sistema
* Visualizar el orden jerárquico de los cargos
* Visualizar en forma de árbol desplegable los usuarios respecto a al cargo seleccionado
  1. Usuarios.
* Visualizar los usuarios registrados en el sistema
* Visualizar la descripción a cerca del usuario seleccionado.
  1. Tipos de solicitud
* Consultar los tipos de solicitud que hay disponibles en el sistema con su determinada información.

Módulo 2: Flujo de solicitudes.

1. Administrador

El aplicativo permitirá

* 1. Definir la ruta que tendrá el envió de cada solicitud dependiendo de su tipo
  2. Modificar la ruta de envió de cada solicitud

1. Usuario

El aplicativo permitirá

* 1. Visualizar la ruta que tendrá la solicitud previamente enviada
  2. Visualizar el destino final que tendrá la solicitud previamente enviada

Módulo 3: Gestión de solicitudes.

1. Administrador

El aplicativo permitirá

* 1. Visualizar solicitudes
  2. Cambiar el diseño visual del perfil de los usuarios cuando sea necesario

1. Usuario

El aplicativo permitirá

* 1. Visualizar todas sus solicitudes o tramites
  2. Crear solicitud
  3. Visualizar notificaciones a cerca de las solicitudes que tiene pendientes
  4. Visualizar el estado de su trámite o solicitud (aprobado, no aprobado o en proceso)
  5. Visualizar en que dependencia se encuentra su trámite o solicitud
  6. Aceptar solicitudes o tramites (En caso de que pueda hacerlo)
  7. Rechazar solicitudes o tramites (En caso de que pueda hacerlo)
  8. REQUERIMIENTOS DEL DESARROLLO
     1. Recursos humanos
* Personal para el desarrollo del software
  + 1. Recursos físicos
* Computadores

# HITOS – ENTREGABLES Y FECHAS DE ENTREGA

|  |  |  |
| --- | --- | --- |
| Nro. | Entregable | Fecha de entrega |
| 1 | Modulo estructura organizacional y tipos de solicitud |  |
| 2 | Modulo flujo de solicitudes |  |
| 3 | Modulo gestión de solicitudes |  |
| 4 | Modulo creación de formatos para cada tipo de solicitud |

**Sprint 2.**

**Modulo 1: Estructura organizacional.**

Fase 1. Historias de usuario.

Administrador.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ENUNCIADO DE LA HISTORIA** | | | | |
| **ID** | Rol | Característica / Funcionalidad | Razón / Resultado | |
| 1 | Como administrador | Necesito crear dependencias nuevas | Para registrar nuevos actores principales al sistema | |
| **CRITERIOS DE ACEPTACION** | | | | |
| # | Criterio de aceptación (Titulo) | Contexto | Evento | Resultado / Comportamiento esperado |
| 1 | Dependencia no creada | Dependencia sin haber sido creada previamente | Cuando se despliegue un panel de dialogo tipo mensaje | Se mostrará un mensaje positivo respecto a la inserción de una nueva dependencia |
| 2 | Dependencia ya creada | Dependencia creada previamente con datos únicos iguales a la dependencia por crearse | Cuando se despliegue un panel de dialogo tipo mensaje | Se mostrará un mensaje negativo respecto a la intención de insertar una nueva dependencia |

Tabla . Historia de usuario 1 (Administrador)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ENUNCIADO DE LA HISTORIA** | | | | |
| **ID** | Rol | Característica / Funcionalidad | Razón / Resultado | |
| 2 | Como administrador | Necesito actualizar las dependencias registradas | Para modificar datos de una dependencia especifica cada que sea necesario | |
| **CRITERIOS DE ACEPTACION** | | | | |
| # | Criterio de aceptación (Titulo) | Contexto | Evento | Resultado / Comportamiento esperado |
| 1 | Dependencia existente | Dependencia registrada en el sistema previamente | Cuando se despliegue un panel de dialogo tipo mensaje | Se mostrará un mensaje positivo respecto a la modificación de datos de la dependencia actualizada |
| 2 | Dependencia no existente | dependencia no registrada en el sistema previamente | Cuando se despliegue un panel de dialogo tipo mensaje | Se mostrará un mensaje negativo respecto a la dependencia que quiere modificar |

Tabla . Historia de usuario 2 (Administrador)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ENUNCIADO DE LA HISTORIA** | | | | |
| **ID** | Rol | Característica / Funcionalidad | Razón / Resultado | |
| 3 | Como administrador | Necesito eliminar dependencias | Para eliminar una dependencia especifica cada que sea necesario | |
| **CRITERIOS DE ACEPTACION** | | | | |
| # | Criterio de aceptación (Titulo) | Contexto | Evento | Resultado / Comportamiento esperado |
| 1 | Dependencia existente | Dependencia registrada en el sistema previamente | Cuando se despliegue un panel de dialogo tipo mensaje | Se mostrará un mensaje positivo respecto a la eliminación de la dependencia seleccionada |
| 2 | Dependencia no existente | dependencia no registrada en el sistema previamente | Cuando se despliegue un panel de dialogo tipo mensaje | Se mostrará un mensaje negativo respecto a la dependencia que quiere eliminar |

Tabla . Historia de usuario 3 (Administrador)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ENUNCIADO DE LA HISTORIA** | | | | |
| **ID** | Rol | característica / Funcionalidad | Razón / Resultado | |
| 4 | Como administrador | Necesito ingresar orden jerárquico de las dependencias | Para permitir el reconocimiento automático del orden que debe seguir el flujo de cada solicitud | |
| **CRITERIOS DE ACEPTACION** | | | | |
| # | Criterio de aceptación (Titulo) | Contexto | Evento | Resultado / Comportamiento esperado |
| 1 | Orden valido | Orden que concuerda con el grado o nivel de jerarquía de todas las dependencias | Cuando se despliegue un panel de dialogo tipo mensaje | se mostrará un mensaje positivo ante la adición de un nuevo orden jerárquico de las dependencias |
| 2 | Orden no valido | orden que no concuerda con el nivel de jerarquía de las dependencias | Cuando se despliegue un panel de dialogo tipo mensaje | Se mostrará un mensaje negativo ante la intención de adicionar un nuevo orden jerárquico en las dependencias |

Tabla . Historia de usuario 4 (Administrador)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ENUNCIADO DE LA HISTORIA** | | | | |
| **ID** | Rol | característica / Funcionalidad | Razón / Resultado | |
| 5 | Como administrador | Necesito modificar orden jerárquico de las dependencias | Para permitir tener el flujo de las solicitudes actualizadas | |
| **CRITERIOS DE ACEPTACION** | | | | |
| # | Criterio de aceptación (Titulo) | Contexto | Evento | Resultado / Comportamiento esperado |
| 1 | Orden valido | Orden que concuerda con el grado o nivel de jerarquía de todas las dependencias | Cuando se despliegue un panel de dialogo tipo mensaje | se mostrará un mensaje positivo ante la adición de un nuevo orden jerárquico de las dependencias |
| 2 | Orden no valido | orden que no concuerda con el nivel de jerarquía de las dependencias | Cuando se despliegue un panel de dialogo tipo mensaje | Se mostrará un mensaje negativo ante la intención de adicionar un nuevo orden jerárquico en las dependencias |

Tabla . Historia de usuarios 5 (Administrador)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ENUNCIADO DE LA HISTORIA** | | | | |
| **ID** | Rol | característica / Funcionalidad | Razón / Resultado | |
| 6 | Como administrador | Necesito visualizar las dependencias registradas en el sistema | Para permitir consultar los datos que tiene cada dependencia seleccionada | |
| **CRITERIOS DE ACEPTACION** | | | | |
| # | Criterio de aceptación (Titulo) | Contexto | Evento | Resultado / Comportamiento esperado |
| - |  |  |  |  |
| - |  |  |  |  |

Tabla . Historia de usuario 6 (Administrador)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ENUNCIADO DE LA HISTORIA** | | | | |
| **ID** | Rol | característica / Funcionalidad | Razón / Resultado | |
| 7 | Como administrador | Necesito crear nuevos usuarios | Para permitir la interacción de los administrativos con el sistema | |
| **CRITERIOS DE ACEPTACION** | | | | |
| # | Criterio de aceptación (Titulo) | Contexto | Evento | Resultado / Comportamiento esperado |
| 1 | Usuario no existente | usuario no registrado previamente en el sistema | Cuando se despliegue un panel de dialogo tipo mensaje | se mostrará un mensaje positivo respecto a la creación del nuevo usuario |
| 2 | Usuario existente | usuario registrado previamente en el sistema con los mismos datos únicos (documento, código) | Cuando se despliegue un panel de dialogo tipo mensaje | se mostrará un mensaje negativo respecto a la intención de crear un nuevo usuario |

Tabla . Historia de usuario 7 (Administrador)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ENUNCIADO DE LA HISTORIA** | | | | |
| **ID** | Rol | característica / Funcionalidad | Razón / Resultado | |
| 8 | Como administrador | necesito enlazar los cargos con los usuarios ya creados | para permitir unir datos e ir consolidando una base de datos ordenada y completa | |
| **CRITERIOS DE ACEPTACION** | | | | |
| # | Criterio de aceptación (Titulo) | Contexto | Evento | Resultado / Comportamiento esperado |
| - |  |  |  |  |
| - |  |  |  |  |

Tabla . Historia de usuario 8 (Administrador)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ENUNCIADO DE LA HISTORIA** | | | | |
| **ID** | Rol | característica / Funcionalidad | Razón / Resultado | |
| 9 | Como administrador | necesito modificar usuarios | para permitirle al sistema tener información actualizada | |
| **CRITERIOS DE ACEPTACION** | | | | |
| # | Criterio de aceptación (Titulo) | Contexto | Evento | Resultado / Comportamiento esperado |
| 1 | usuario existente | usuario si existe dentro de la base de datos | Cuando se despliegue un panel de dialogo tipo mensaje | se mostrará los campos actualizados |
| 2 | usuarios no existentes | usuario no está registrado en la base de datos | Cuando se despliegue un panel de dialogo tipo mensaje | se mostrará un mensaje negativo indicando que el usuario no existe |

Tabla . Historia de usuario 9 (Administrador)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ENUNCIADO DE LA HISTORIA** | | | | |
| **ID** | Rol | característica / Funcionalidad | Razón / Resultado | |
| 10 | Como administrador | necesito eliminar usuarios | para permitirle al sistema borrar usuarios registrados previamente cada que sea necesario | |
| **CRITERIOS DE ACEPTACION** | | | | |
| # | Criterio de aceptación (Titulo) | Contexto | Evento | Resultado / Comportamiento esperado |
| 1 | usuario existente | usuario registrado previamente en la base de datos | Cuando se despliegue un panel de dialogo tipo mensaje | se mostrará un mensaje exitoso indicando que se ha eliminado el usuario |
| 2 | Usuario no existente | usuario que no existe dentro del sistema | Cuando se despliegue un panel de dialogo tipo mensaje | se mostrará un mensaje negativo indicando que el usuario no existe |

Tabla . Historia de usuario 10 (Administrador)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ENUNCIADO DE LA HISTORIA** | | | | |
| **ID** | Rol | característica / Funcionalidad | Razón / Resultado | |
| 11 | Como administrador | necesito visualizar usuarios | para validar datos de los usuarios | |
| **CRITERIOS DE ACEPTACION** | | | | |
| # | Criterio de aceptación (Titulo) | Contexto | Evento | Resultado / Comportamiento esperado |
| - |  |  |  |  |
| - |  |  |  |  |

Tabla . Historia de usuario 11 (Administrador)

Fase 2. Planeación

1. **Plan proyecto del módulo.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Nombre de la tarea | Responsable | Fecha de inicio | Fecha final | Días | Estado |
| 1 | **Historias de usuario** | Camilo Soto | 2/7 | 2/14 | 7 | Finalizado |
| 1.2 | **Definir requerimientos** | Camilo Soto | 2/7 | 2/9 | 2 | Finalizado |
| 1.3 | **Definir tipos de usuario** | Camilo Soto | 2/9 | 2/14 | 5 | Finalizado |
| 2 | **Planeación** | Camilo Soto | 2/14 | 2/14 | 0 | Finalizado |
| 2.1 | **Project plan** | Camilo Soto | **2/14** | **2/14** | **0** | Finalizado |
| 2.2 | **Análisis** | Camilo Soto | **2/14** | **2/15** | **1** | Finalizado |
| 2.2.1 | Casos de uso | Camilo Soto | 2/14 | 2/15 | 1 | Finalizado |
| 2.3 | **Diseño** | Camilo Soto | **2/15** | **2/29** | **14** | Finalizado |
| 2.3.1 | Modelo entidad - relación | Camilo Soto | 2/20 | 2/22 | 2 | Finalizado |
| 2.3.2 | Diagrama de actividades | Camilo Soto | 2/22 | 2/29 | 7 | Finalizado |
| 3 | **Desarrollo** | Camilo Soto | 2/29 | 4/12 | 43 | En progreso |
| 3.1 | **Back-end** | Camilo Soto | 2/29 | 3/26 | 26 | Finalizado |
| 3.2 | **Front-end** | Camilo Soto | 3/26 | 4/10 | 15 | En progreso |
| 3.3 | **Pruebas** | Camilo Soto | 4/10 | 4/12 | 2 | Sin empezar |

Tabla 2. Plan project

1. **Análisis.**

# **Descripción de Actores**

## Jefe de área o dependencia.

El empleado que cumple con el rol de jefe dentro de un área o departamento en la Universidad de Manizales es aquel responsable de la administración de dicha área, de que el trabajo este bien realizado en tiempo y forma de acuerdo a las necesidades y enfoque de la empresa, su principal labor es liderar todo el equipo de trabajo guiando todo el esfuerzo colectivo a la búsqueda de la solución de problemas y cumplimiento de objetivos.

Un jefe de área en la Universidad de Manizales a parte de cumplir con las anteriores funciones también debe responsabilizarse de los tiempos de los tiempos de cada uno de los empleados que laboran dentro del departamento a su cargo.

El rol del jefe de área dentro del aplicativo es determinante para el funcionamiento del mismo, pues es el tipo de usuario que va a tener ciertos derechos de privilegio en el manejo del aplicativo como por ejemplo poder aprobar en primera instancia las solicitudes concernientes a los empleados de su área, siendo así determinante en los módulos del software que enfatizan en el flujo de las solicitudes.

1. Empleado.

El empleado es un tipo de usuario administrativo de la empresa al igual pero el usuario mencionado anteriormente (jefe de área) solo que a diferencia de él este usuario no es jefe de ningún área y cumple con un rol menor a nivel jerárquico dentro de la institución. Si bien este usuario no debe cumplir con el rol de ser líder dentro de un área, de igual manera debe cumplir a cabalidad con sus responsabilidades aportando de esta manera al cumplimiento de objetivos en primera instancia dentro de la dependencia donde labora y así mismo, objetivos mas generales a nivel de empresa.

El rol del empleado raso dentro del aplicativo no es relevante en cuanto a gestión de terceros, pues la única gestión que podrá hacer dentro del aplicativo será gestionar sus propias solicitudes, crearlas, enviarlas y consultar las mismas, de esta manera no se considera la opción de otorgar derechos administrativos a este usuario.

1. Administrador.

El administrador del aplicativo es el tipo de usuario otorgado al empleado de la Universidad de Manizales que cumple con las labores respectivas de mantenimiento del sistema virtual (SIGUM), dichos administrativos que cumplen con la función estan laborando dentro de la dependencia de tecnologías de la información, este empleado cumple generalmente con actividades como cubrimiento de aspectos de seguridad de la pagina web, monitorizar el estado de sitio web, mantener actualizado el entorno del aplicativo, etc.

El rol del administrador dentro del aplicativo será vital para la alimentación de la base de datos y constante actualización de la misma, dicho usuario tendrá permisos ilimitados para la gestión absoluta del sistema, destacando también que como empleado de la Universidad de Manizales no solo cumple con el papel de gestionar y mantener el aplicativo, sino que también al igual que el resto de empleados podrá crear solicitudes y enviarlas.

* 1. **Descripción de casos de uso.**

## Gestionar registros del aplicativo

|  |  |  |
| --- | --- | --- |
| Caso de Uso | Gestionar registros del aplicativo | Identificador:  CU-1 |
| Actores | Administrador | |
| Tipo | Primario | |
| Referencias | REQ-1: Crear dependencias nuevas  REQ-2: Actualizar dependencias  REQ-3: Eliminar dependencias  REQ-4: Ingresar orden jerárquico de las dependencias  REQ-5: Modificar orden jerárquico de las dependencias  REQ-6: Visualizar las dependencias registradas en el sistema  REQ-7: Visualizar el orden jerárquico de las dependencias  REQ-8: Visualizar la descripción a cerca de la dependencia seleccionada  REQ-9: Visualizar en forma de árbol desplegable los tipos de usuario pertenecientes a las dependencias  REQ-10: Crear nuevos cargos  REQ-11: Asignar cargos que contiene la dependencia  REQ-12: Actualizar cargos  REQ-13: Eliminar cargos  REQ-14: Ingresar orden jerárquico de los cargos  REQ-15: Modificar orden jerárquico de los cargos  REQ-16: Visualizar los cargos registrados en el sistema  REQ-17: Visualizar el orden jerárquico de los cargos  REQ-18: Visualizar la descripción a cerca de los cargos seleccionados  REQ-19: Visualizar en forma de árbol desplegable las personas pertenecientes a dicho cargo  REQ-20: Crear usuarios nuevos  REQ-21: Asignar cargos a cada empleado  REQ-22: Actualizar usuarios  REQ-23: Eliminar usuarios  REQ-24: Modificar orden jerárquico a los usuarios  REQ-25: Visualizar los usuarios registrados en el sistema  REQ-26: Visualizar el orden jerárquico de los usuarios  REQ-27: Visualizar la descripción a cerca de los usuarios seleccionados  REQ-28: Crear tipo de solicitud  REQ-29: Modificar tipo de solicitud  REQ-30: Eliminar tipo de solicitud | |
| Precondición | Ingresar como administrador | |
| Postcondición | Gestión de los registros del aplicativo | |
| Descripción | Por medio de este caso se permitirá el administrador realizar las acciones básicas (crear, modificar, eliminar, consultar) para registrar nueva información contenida en el aplicativo. | |
| Resumen | Este caso es la primero que debe desarrollar el administrador del subsistema para poblar el subsistema con las dependencias respectivas. | |

Tabla 19. Descripción de caso - gestionar registros del aplicativo

Diagrama.

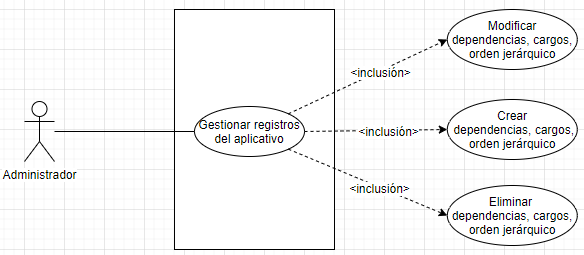


Ilustración 1. Diagrama caso de uso - gestión de registros del aplicativo

## Consultar usuarios.

|  |  |  |
| --- | --- | --- |
| Caso de Uso | Consultar información de usuarios | Identificador:  CU-2 |
| Actores | Jefe de dependencia  Empleado | |
| Tipo | Secundario | |
| Referencias | REQ-36: Visualizar los usuarios registrados en el sistema  REQ-37: Visualizar la descripción a cerca del usuario seleccionado. | |
| Precondición | Iniciar sesión en el sistema | |
| Postcondición | Información visualizada del usuario seleccionado | |
| Descripción | En este caso se permitirá cualquier usuario consultar y visualizar la información pública de cualquier usuario que desee. | |
| Resumen | Este caso se debe desarrollar por medio del administrador del subsistema para poblar el subsistema con los cargos respectivos para los usuarios de cada dependencia. | |

Tabla 21. Descripción de caso - gestionar usuario

## Diagrama.

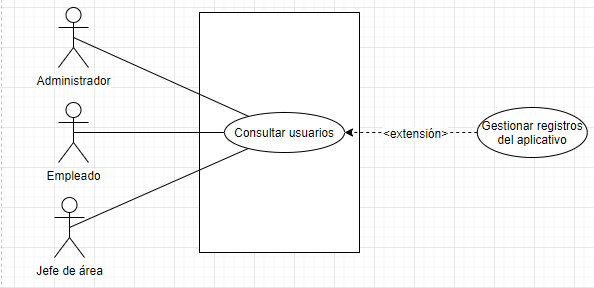


Ilustración . Diagrama caso de uso - consultar usuarios.

## Consultar dependencias.

|  |  |  |
| --- | --- | --- |
| Caso de Uso | Consultar dependencias | Identificador:  CU-3 |
| Actores | Jefe de dependencia  Empleado | |
| Tipo | Secundario | |
| Referencias | REQ-28: Visualizar las dependencias registradas en el sistema  REQ-29: Visualizar el orden jerárquico de las dependencias  REQ-30: Visualizar la descripción a cerca de la dependencia seleccionada  REQ-31: Visualizar en forma de árbol desplegable los tipos de usuario pertenecientes a las dependencias | |
| Precondición | Iniciar sesión en el sistema | |
| Postcondición | Información consultada de la dependencia | |
| Descripción | Este caso reflejará las acciones de consulta que tendrán los usuarios hacia las dependencias registrada en el aplicativo | |
| Resumen | Esta funcionalidad será opcional para cualquier usuario registrado en el sistema la cual permitirá mostrar al usuario las dependencias existentes y su información correspondiente. | |

Tabla 23. Descripción del caso - consultar dependencias

Diagrama.

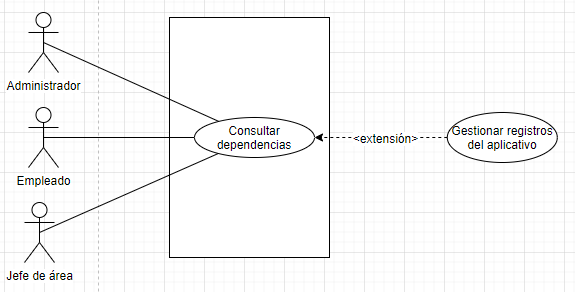


Ilustración . Diagrama caso de uso - consultar dependencias.

## Consultar cargos.

|  |  |  |
| --- | --- | --- |
| Caso de Uso | Consultar cargos | Identificador:  CU-4 |
| Actores | Jefe de dependencia  Empleado | |
| Tipo | Secundario | |
| Referencias | REQ-32: Visualizar cargos  REQ-33: Visualizar los cargos registrados en el sistema  REQ-34: Visualizar el orden jerárquico de los cargos  REQ-35: Visualizar en forma de árbol desplegable los usuarios respecto a al cargo seleccionado | |
| Precondición | Iniciar sesión en el sistema | |
| Postcondición | Información consultada de los cargos | |
| Descripción | Este caso reflejará las acciones de consulta que tendrán los usuarios hacia los cargos registrada en el aplicativo | |
| Resumen | Esta funcionalidad será opcional para cualquier usuario registrado en el sistema la cual permitirá mostrar al usuario las dependencias existentes y su información correspondiente. | |

Tabla 25. Descripción del caso - consultar cargos

Diagrama.

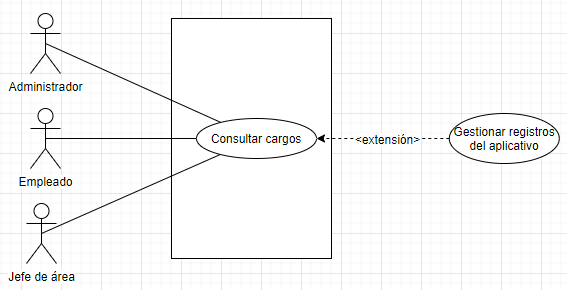


Ilustración . Diagrama caso de uso - consultar cargos.

**3. Diseño**

* 1. **Diagrama de actividades.**

1. Gestionar registros.

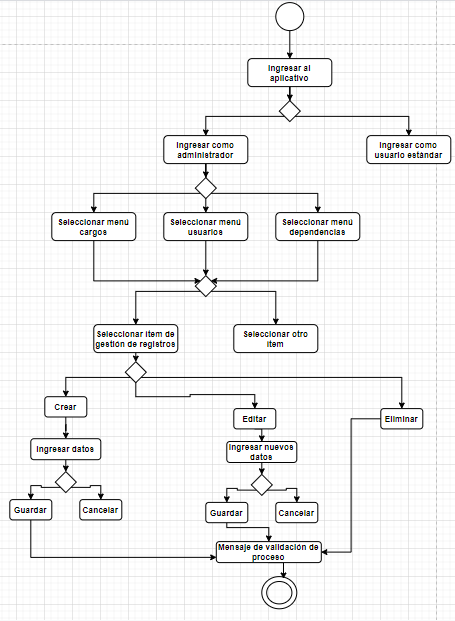


Ilustración . Gestionar registros del aplicativo

1. Consultar cargos.

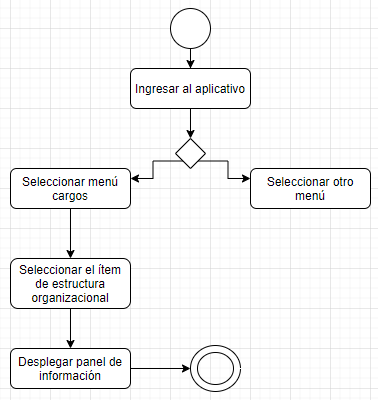


Ilustración 6. Actividad consultar cargos

1. Consultar dependencias.

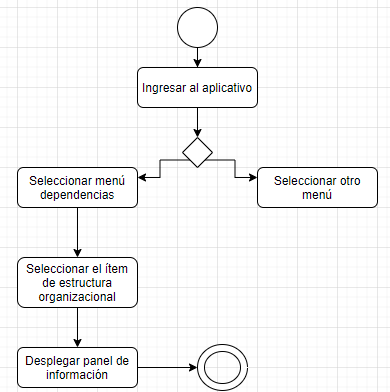


Ilustración 7. Actividad consultar dependencias

1. Consultar usuarios.

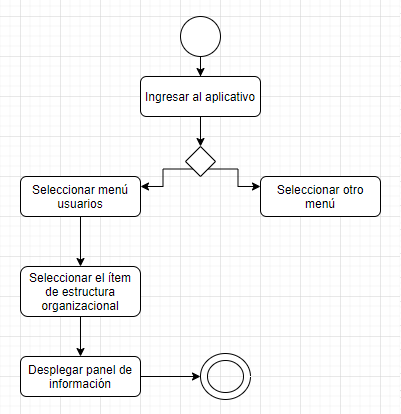


Ilustración 8. Actividad consultar usuarios.

* 1. **Modelo entidad – relación.**

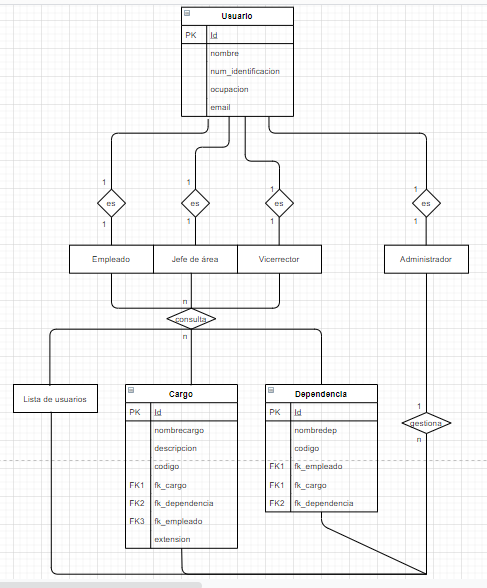


Ilustración 9. Modelo entidad – relación.

* 1. **Diccionario de datos.**

1. Empleados.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributos** | | | | | |
| **Restricción** | **Nombre** | **Descripción** | **Tipo** | **Tamaño** | **Requerido** |
| Llave primaria | id | Se utiliza para identificar inequívocamente cada fila | integer | - | si |
| No tiene | nombre | Es el nombre propio que posee el empleado registrado | character varying | 255 | si |
| No tiene | num\_identificacion | Es el número que posee el documento de identidad del empleado registrado | integer | - | si |
| No tiene | ocupación | El nombre de la labor que cumple el empleado dentro de la institución | character | 255 | no |
| No tiene | email | Correo electrónico que posee el empleado | character varying | 255 | si |
| No tiene | createdAt | Fecha automática otorgada justo cuando se creó el registro de dicho empleado | date | - | no |
| No tiene | updatedAt | Fecha automática otorgada justo cuando se actualizo por última vez alguno de los campos del empleado | date | - | no |

Tabla 6. Atributos de 'empleados'

1. Dependencias.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributos** | | | | | |
| **Restriccion** | **Nombre** | **Descripción** | **Tipo** | **Tamaño** | **Requerido** |
| Llave primaria | id | Se utiliza para identificar inequívocamente cada fila | integer | - | si |
| No tiene | nombre | Es el nombre propio que posee cada dependencia registrado | character varying | 255 | si |
| No tiene | código | Valor identificador numérico único que se le da a la dependencia de cierto número de caracteres | bigint | - | si |
| Llave foránea | fk\_empleado | Enlace directo a la tabla empleado para saber que usuario es jefe de dicha área | bigint | - | si |
| Llave foránea | fk\_dependencia | Enlace directo a la misma tabla para saber si dicha dependencia depende de otra dependencia diferente | bigint | - | no |
| Llave foránea | fk\_cargo | Enlace directo a la tabla cargos para saber que cargos posee la dependencia | bigint | - | si |
| No tiene | createdAt | Fecha automática otorgada justo cuando se creó el registro de dicho empleado | date | - | no |
| No tiene | updatedAt | Fecha automática otorgada justo cuando se actualizo por última vez alguno de los campos del empleado | date | - | no |

Tabla 27. Atributos de 'dependencias'

1. Cargos.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributos** | | | | | |
| **Restricción** | **Nombre** | **Descripción** | **Tipo** | **Tamaño** | **Requerido** |
| Llave primaria | id | Se utiliza para identificar inequívocamente cada fila | integer | - | si |
| No tiene | nombre | Es el nombre propio que posee cada cargo registrado | character varying | 255 | si |
| No tiene | código | Valor identificador numérico único que se le da al cargo de cierto número de caracteres | bigint | - | si |
| No tiene | Descripción | Para indicar que función tiene dicho cargo dentro de la dependencia | character varying | 255 | no |
| No tiene | Extensión | Número de extensión telefónica que posee el cargo | integer | - | si |
| Llave foránea | fk\_empleado | Enlace directo a la tabla empleado para saber que usuario ocupa dicho cargo | bigint | - | si |
| Llave foránea | fk\_dependencia | Enlace directo a la tabla dependencia para saber a qué dependencia pertenece el cargo | bigint | - | si |
| Llave foránea | fk\_cargo | Enlace directo a la misma tabla para saber si hay un cargo jefe | bigint | - | no |
| No tiene | createdAt | Fecha automática otorgada justo cuando se creó el registro de dicho empleado | date | - | no |
| No tiene | updatedAt | Fecha automática otorgada justo cuando se actualizo por última vez alguno de los campos del empleado | date | - | no |

Tabla 28. Atributos de 'cargos'

# WEBGRAFIA

GESTION.ORG. (2018). *GESTION.ORG*. Obtenido de GESTION.ORG: https://www.gestion.org/que-son-los-metadatos/

Pinto, C. A. (julio de 2007). *Diseño y desarrollo de flujos de trabajo y formularios electronicos para la automatizacion de procesos de solicitudes basados en documentos.*

PROFIT LTDA, W. (2020). *GESTIONPOLIS*. Obtenido de GESTIONPOLIS: https://www.gestiopolis.com/la-estructura-organizacional/

Universidad de Manizales. (2017). *Universidad de Manizales*. Obtenido de Universidad de Manizales: https://umanizales.edu.co/

1. Universidad de Manizales. (2017). *Universidad de Manizales*. Obtenido de Universidad de Manizales: https://umanizales.edu.co/ [↑](#footnote-ref-1)
2. *sinnaps*. (2019). Obtenido de sinnaps: https://www.sinnaps.com/blog-gestion-proyectos/metodologia-scrum [↑](#footnote-ref-2)
3. Pinto, C. A. (julio de 2007). *Diseño y desarrollo de flujos de trabajo y formularios electronicos para la automatizacion de procesos de solicitudes basados en documentos.* [↑](#footnote-ref-3)
4. *sinnaps*. (2019). Obtenido de sinnaps: https://www.sinnaps.com/blog-gestion-proyectos/metodologia-scrum [↑](#footnote-ref-4)
5. Conover, R. (s.f.). *Nodejs*. Obtenido de Nodejs: https://nodejs.org/es/ [↑](#footnote-ref-5)
6. contributors, M. a. (2020). *MDN web docs*. Obtenido de MDN web docs: https://developer.mozilla.org/es/docs/Web/JavaScript [↑](#footnote-ref-6)
7. Google. (s.f.). *Google chrome*. Obtenido de Google chrome: https://www.google.com.mx/chrome/?brand=CHBD&gclsrc=aw.ds&&gclid=EAIaIQobChMI9\_LChsmN6AIVC5-fCh3aIg-PEAAYASAAEgK\_TPD\_BwE [↑](#footnote-ref-7)
8. Twisted. (s.f.). *TwistedMatrix*. Obtenido de TwistedMatrix: https://twistedmatrix.com/trac/ [↑](#footnote-ref-8)
9. Bottstrap. (s.f.). *Bootstrap*. Obtenido de Bootstrap: https://getbootstrap.com/

   dmatrix.com/trac/ [↑](#footnote-ref-9)
10. Fundation, j. (2020). *jQuery*. Obtenido de jQuery: https://jquery.com/

    [↑](#footnote-ref-10)