

**Sistema de información para el registro y gestión de vendedores informales de los
municipios de Colombia**

Docentes Asesores:

Dianalin Neme Prada

Iván Romero Flórez

Ing. Luis Yeferson Reyes

Ing. Jorge Iván Lozano

Mayo 2021

UNIVERSIDAD ANTONIO NARIÑO.

FACULTAD DE INGENIERÍA DE SISTEMAS.

ESPECIALIZACION EN INGENIERIA DE SOFTWARE

Tabla de Contenidos

RESUMEN	
INTRODUCCION	1
1. FORMULACIÓN DEL PROBLEMA.....	2
2. OBJETIVOS	4
2.1 Objetivo General.....	4
2.2 Objetivos Específicos.....	4
3. MARCO DE REFERENCIA	5
3.1 Estado del Arte.....	5
3.2 Impacto	6
3.3 Componente de Innovación	7
4. MARCO TEÓRICO.....	9
5. METODOLOGÍA.....	15
6. PROCESO DE SOFTWARE.....	20
6.1 Requerimientos Funcionales.....	20
6.2 Requerimientos No Funcionales	21
6.3 Diseño y Arquitectura	26
6.4 Diagrama de despliegue.....	29
6.5 Caso de Uso Arquitecturalmente Relevante	30
6.6 Diagrama de Secuencia.....	34
6.7 Diagrama de Clases.....	37
6.8 Arquitectura de Alto Nivel.....	39
7. CONSTRUCCIÓN	40
8. PRUEBAS.....	44
8.1 Escenario de calidad adecuación funcional	44
9. INSTALACION Y CONFIGURACION.....	52
CONCLUSIONES	62
REFERENCIAS.....	64

Listado de Tablas

Tabla 1 - Requerimiento funciona Registro de Vendedores – Fuente propia.....	20
Tabla 2 - Requerimiento funcional Actualizar / Eliminar Vendedor Informal - Fuente propia ...	21
Tabla 3 - Requerimiento funcional Generar Reportes - Fuente propia.....	21
Tabla 4 - Requerimiento no funcional Seguridad usuario Encuestador - Fuente propia	22
Tabla 5 - Requerimiento no funcional Seguridad usuario Administrador - Fuente propia	23
Tabla 6 - Requerimiento no funcional Seguridad otros usuarios - Fuente propia	23
Tabla 7 - Requerimiento no funcional Disponibilidad - Fuente propia	24
Tabla 8 - Requerimiento no funcional Interoperabilidad - Fuente propia	25
Tabla 9 - Requerimiento no funcional Usabilidad - Fuente propia	25
Tabla 10 - Requerimiento no funcional Escalabilidad - Fuente propia	26
Tabla 11 - Caso de uso Registrar Vendedor - Fuente propia.....	32
Tabla 12 - Caso de uso Consultar Vendedor - Fuente propia.....	32
Tabla 13 - Caso de uso Actualizar Vendedor - Fuente propia.....	33
Tabla 14 - Caso de uso Eliminar Vendedor - Fuente propia.....	33
Tabla 15 - Caso de uso Generar Reportes - Fuente propia	34
Tabla 16 - Caso de Prueba Crear Vendedor.....	45
Tabla 17 - Caso de Prueba Actualizar Vendedor.....	46
Tabla 18 - Caso de Prueba Eliminar Vendedor	47
Tabla 19 - Caso de Prueba Consultar Vendedor.....	48
Tabla 20 - Caso de Prueba Generar Reporte.....	49
Tabla 21 - Escenario de Calidad para atributo de Mantenibilidad.....	50

Listado de Imágenes

Ilustración 1 - tablero KANBAN Proyecto sistema de información para el registro y gestión de vendedores informales en los municipios de Colombia – Imagen tomada de tablero creado en herramienta TRELLO (https://trello.com/es).....	16
Ilustración 2 - Cronograma para el desarrollo del sistema de información para el registro y gestión de vendedores informales en los municipios de Colombia – Fuente propia	19
Ilustración 3- Modelo-Vista-Controlador - Fuente propia.....	27
Ilustración 4 - Diagrama de Despliegue - Fuente propia	29
Ilustración 5 - Caso de Uso Arquitecturalmente Relevante - Fuente propia	31
Ilustración 6 - Diagrama de secuencia Registrar Vendedor - Fuente propia	35
Ilustración 7 - Diagrama de secuencia Consultar Vendedor - Fuente propia	35
Ilustración 8 - Diagrama de secuencia Actualizar Vendedor - Fuente propia	36
Ilustración 9 - Diagrama de secuencia Eliminar Vendedor - Fuente propia.....	36
Ilustración 10 - Diagrama de secuencia Generar Reportes - Fuente propia	37
Ilustración 11 - Diagrama de Clases - Fuente propia.....	38
Ilustración 12 - Diagrama de Arquitectura de Alto Nivel - Fuente propia	39
Ilustración 13 - Entorno de Desarrollo Visual Studio Code – Fuente Propia.....	41
Ilustración 14 - Entorno de desarrollo, uso de MVC – Fuente Propia.....	41
Ilustración 15 - Pantalla de Ingreso S.I. Vendedores Informales - Fuente propia.	42
Ilustración 16 - Formulario de registro de un vendedor informal - Fuente propia	43
Ilustración 17 - Integración de Sonar Lint a Visual Studio Code (Entorno de Desarrollo) – Fuente propia	51
Ilustración 18 - Errores de codificación detectados por Sonar Lint - Fuente propia	51
Ilustración 19 - Formulario para la creación de usuario administrador y base de datos de Odoo – Fuente propia	54
Ilustración 20 - Pantalla para ingreso a Odoo - Fuente propia	56
Ilustración 21 - Pantalla de visualización de aplicaciones asociadas al usuario administrador - Fuente propia	56
Ilustración 22 - Opción para búsqueda de aplicaciones ya existentes en Odoo o creadas.....	57
Ilustración 23 - Lista de aplicaciones existentes y registradas en Odoo.....	57
Ilustración 24 - Actualización de módulos en Odoo - Fuente propia	58
Ilustración 25 - Cargue de nuevas aplicaciones a Odoo - Fuente Propia.....	58
Ilustración 26 - Identificación de aplicación a asignar permisos - Fuente propia	59
Ilustración 27 - Se identifica a que aplicación se debe asignar de permisos a usuarios poder ejecutar la aplicación - Fuente propia	59
Ilustración 28 - Asignación de permisos sobre la aplicación de interés - Fuente propia.....	60
Ilustración 29 - Se valida que la nueva aplicación esta ya cargada en el entorno de Odoo - Fuente propia	60
Ilustración 30 - Validación de acceso a aplicación - Fuente propia	61

RESUMEN

El objetivo de este trabajo es realizar el análisis, diseño e implementación de una aplicación que permita el registro y gestión de la información básica de vendedores informales de los municipios de Colombia, tomando como referencia lo propuesto en la política pública de vendedores informales.

En la primera parte se presenta la situación actual relacionada con la problemática normativa existente para la identificación, categorización y vinculación a la formalidad laboral de los vendedores informales, así como la existencia de sistemas de información que apoyan la gestión de información de los mismos, y realizar la evaluación de diferentes opciones de herramientas existentes en el mercado que cumplen parcial o totalmente las necesidades de vincular a los vendedores informales a proyectos de mejoramiento de sus condiciones sociales y analizar el impacto de la implementación de una solución de software.

Para documentar las necesidades de la solución, se construye una descripción de alto nivel de los requerimientos funcionales, no funcionales y de la arquitectura.

La solución planteada cumple y aprovecha las recomendaciones de las buenas prácticas de arquitecturas de software, iniciando con una definición de las entidades u objetos que representaran a los actores de la aplicación, posteriormente el detalle de los casos de uso que explicaran a alto

nivel las funcionalidades de la solución, y finalmente se detallarán los diagramas que muestran mayor detalle de la misma.

La construcción de la solución se hace utilizando herramientas y tecnologías consideradas actualmente como líderes para el desarrollo de software, que ayudan a lograr reducciones de tiempo en el proceso de codificación y permitiendo la aplicación de la programación orientada a objetos y de buenas prácticas de desarrollo. Finalmente se realizan pruebas de la solución para validar que los objetivos planteados y los requerimientos funcionales y no funcionales han sido cumplidos.

INTRODUCCION

Las ventas informales se han convertido, en los últimos años, en una forma de buscar sustento para muchos de los colombianos que no logran encontrar formas de vincularse a actividades o empleos más “normales” o formales.

La cantidad y variedad de productos y servicios que se ofrecen a través de las ventas informales, y la cantidad aun mayor de personas que derivan su sustento de estas ventas, hace necesario que se pueda contar con mecanismos y/o herramientas que permitan y faciliten la captura y consolidación de información básica esta personas, ya que como se describe en la Política Pública de Vendedores Informes, en la página 50, solo el 17% de los municipios encuestados cuentan con información actualizada al primer semestre del año 2020 y el 42% restante cuenta con alguna información de años anteriores, y solo el 21% está desarrollando alguna herramienta para el registro de los vendedores informales.

Por lo descrito anteriormente, se creará una aplicación web que permita la captura y mantenimiento de la información básica de los vendedores informales de cualquier región de Colombia, y que brinde información homogénea y centralizada, que facilite posteriores ejercicios de caracterización de las condiciones sociales, demográficas y económicas de las ventas informales.

1. FORMULACIÓN DEL PROBLEMA

En los años recientes, el incremento en la concentración de personas en los principales centros urbanos de Colombia, ha llegado a un porcentaje alrededor del 75% (cifra tomada de la política pública de vendedores informales), lo que ha causado así mismo una mayor demanda no solo en servicios como vivienda, transporte, servicios públicos y salud, si no también está demandando la generación de más fuentes de empleo a un ritmo que, los entes de gobierno municipal, departamental y nacional no han podido atender de manera efectiva. Lo anterior es una tendencia que va a seguir al alza en el futuro próximo.

La difícil situación laboral en Colombia ha disparado la generación de fuentes alternativas de trabajo, lo que ha gestado un incremento considerable en la forma de trabajo llamada ventas informales, que para el año 2019, como lo menciona la política pública de vendedores informales en su introducción, estaba medida en un 60.6%. Así mismo, el no contar con herramientas informáticas que permitan a los entes territoriales poder tener una información más precisa, centralizada y de fácil y rápido acceso, sobre la cantidad de personas en la informalidad laboral, así como la ubicación donde realizan dichas actividades, la condición social y las actividades que realizan, no permite a dichos entes, poder tomar decisiones acertadas sobre qué acciones y proyectos sociales deben gestarse, para lograr cada vez más y mejores condiciones laborales para las personas en dicha informalidad, buscando en la medida justa poder llevar a los trabajadores informales a esquemas de formalidad laboral o a esquemas que les garanticen un mínimo vital.

Así mismo, se busca a través de las tecnologías de la información y las comunicaciones, poder brindar una plataforma que sea de utilidad no solo para los entes territoriales, sino también para los trabajadores informales, en conocer la situación de estos al interior de un municipio o departamento y poder apoyar iniciativas de mejora en las condiciones laborales.

De acuerdo a lo descrito anteriormente, con este proyecto se pretende construir una herramienta que permita y facilite el registro de información básica de los vendedores informales de un municipio, así como su caracterización a través de capturar sus datos básicos, el tipo de productos o servicios que ofrece, la cantidad de personas que dependen de la actividad informal y la ubicación geográfica del sitio donde normalmente el vendedor realiza su actividad.

2. OBJETIVOS

2.1 Objetivo General

Desarrollar una aplicación web que permita el registro de información, análisis básico de datos y georreferenciación de los vendedores informales en cualquier lugar de Colombia.

2.2 Objetivos Específicos

- Gestionar información detallada de los vendedores informales, cumpliendo con las normas y políticas de protección y manejo de datos dictadas por el gobierno nacional (ley 1581 de 2012).
- Permitir realizar consultas sobre las condiciones socio económicas actuales de los vendedores informales registrados.
- Ofrecer una herramienta tecnológica que permita cubrir las necesidades de información básica sobre vendedores informales, a los municipios que, según la política pública de vendedores informales, no cuentan con este tipo de herramientas.
- Definir la línea base de arquitectura lógica y tecnológica que permita cumplir con los requerimientos funcionales y no funcionales, así como los lineamientos de seguridad para la protección de los datos de los vendedores informales.
- Definir escenarios de calidad, asociados a los atributos de calidad de Mantenibilidad y de adecuación funcional.

3. MARCO DE REFERENCIA

3.1 Estado del Arte

Actualmente solo se encuentran a nivel público o privado pocas soluciones informáticas enfocadas puntualmente en el registro de la información de vendedores informales para vincularlos a proyectos de capacitación en habilidades básicas en sistemas a través del SENA o registrarse para recibir ayudas por parte del gobierno por la situación de pandemia. Entre estos sistemas de información tenemos:

1. **Registro Individual de Vendedores (RIVI):** Solo consiste en un formulario web para el registro de información básica de los vendedores informales para recibir ayudas del gobierno por la situación de pandemia.
2. **Instituto para la economía social IPES:** Solo consiste en un formulario web para el registro de información básica de los vendedores informales para recibir ayudas del gobierno por la situación de pandemia, acceder a cursos de formación a través del SENA, acceso a oportunidades laborales o participar en procesos de formalización de la actividad comercial que estén realizando.
3. **Mercadolocal:** plataforma que funciona como un directorio digital. Allí los usuarios encontrarán vendedores de frutas, pescados, verduras, carnes, fritos, de dulces típicos entre otros productos y contarán con la opción de geolocalización para ubicar al vendedor más

cercano, llamarlo y negociar con él. Esta corresponde a la aplicación más aproximada a la solución que se plantea en el presente proyecto.

A nivel internacional se han encontrado 2 posibles soluciones cuyo alcance es poner en contacto a clientes con vendedores ambulantes y que se mencionan a continuación:

1. Aguilar Avilés, C. R. (2018). APP de ventas móviles que permite conectar a clientes con vendedores ambulantes (Tesis de pregrado). Universidad de las Américas, Quito. El siguiente trabajo de titulación tiene como objetivo principal desarrollar una aplicación de ventas móviles que conecte a usuarios con vendedores ambulantes.
2. Proyecto HSP que consiste en una solución para poner en contacto a los vendedores ambulantes que pasan por los pueblos con sus habitantes. A través de una aplicación se puede contactar con los proveedores, observar su catálogo de productos y hacer pedidos. Asimismo, permite geolocalizar a los vendedores para saber cuándo llegan, e incluso enviar notificaciones cuando estén llegando.

3.2 Impacto

Aunque existen bases de datos que permiten el registro de información relacionada con las ventas informales, estas se enfocan principalmente a registrar información general y básica de los vendedores informales, en formularios web básicos tipo Google Forms o Microsoft Forms, cuyo resultado se enfoca al conteo de los mismos y a consolidación en hojas electrónicas. Adicionalmente y como lo expone el documento de la política pública de vendedores informales

en la página 50, columna 2, segundo párrafo, el 73% de los municipios confirmaron que no cuentan con herramientas que permitan la caracterización de las condiciones sociodemográficas y económicas de las ventas informales.

Sin embargo, el proyecto que se plantea corresponde al desarrollo de una aplicación que no solo soportaría el proceso de registro de información y mantenimiento de la misma, sino también su asociación a una ubicación geográfica y tener estructurada la información para facilitar su acceso en cualquier momento. Todo esto permitiría una mejor caracterización de los vendedores y facilitará su posible vinculación a programas sociales de normalización laboral.

3.3 Componente de Innovación

La solución propuesta en el presente proyecto, frente a las soluciones identificadas en el estado del arte ofrecen los siguientes beneficios:

- a. Aplicación WEB que ofrece la posibilidad de vincular más información que solo un formulario de registro de datos básicos.
- b. Asociación de la información básica a una ubicación geográfica que podría permitir la posterior realización de análisis como: área de influencia y concentración de las ventas informales, desplazamientos de los vendedores informales, categorización de los vendedores informales.

- c. La solución aplicará los lineamientos generales en cuanto a la información a registrar y gestionar de la política pública para vendedores informales, política que fue publicada durante el presente año.
- d. Contar con notificaciones vía correo electrónico del registro exitoso en la aplicación, así como de información de beneficios o programas sociales de interés para el vendedor informal.

4. MARCO TEÓRICO

Las ventas informales se han convertido en una alternativa laboral para cientos de personas a nivel nacional. Ante esta situación, es importante poder contar con herramientas que permitan y faciliten el registro de las personas dedicadas a esta actividad, y que a su vez permita facilitar la interacción entre los vendedores informales y los entes territoriales. Hay que entender que las ventas informales son actividades vigiladas y reglamentadas por instituciones públicas de índole nacional, departamental y municipal.

A nivel nacional, la presidencia de la república y el ministerio del trabajo elaboró y publicó la política pública de vendedores informales, la cual está formulada bajo los siguientes preceptos:

1. Estructuración de programas y proyectos encaminados a garantizar el mínimo vital de esta población y el derecho a gozar de una subsistencia en condiciones dignas, implementando alternativas de trabajo formal.
2. Desarrollar programas de capacitación en diversas artes u oficios a través del Servicio Nacional de Aprendizaje.
3. Fomentar proyectos productivos.
4. Reglamentar el funcionamiento de espacios o locales comerciales de interés social, para promover la inclusión y mejorar sus condiciones de vida.

5. Establecer acciones de control y seguimiento que permitan evidenciar la evolución de la situación socioeconómica de la población.
 6. Impulsar investigaciones o estudios sobre los vendedores informales, a fin de enfocar soluciones a sus problemas prioritarios.
 7. Desarrollar un sistema de registro e inscripción de los vendedores informales, el cual permita caracterizarlos para la elaboración de las líneas de acción y programas que integran la política pública.
 8. Disponer de espacios seguros para las actividades que realizan los vendedores informales.
- (Ámbito Jurídico, 2019)

Puntualmente el numeral 7, hace referencia a la necesidad de crear un sistema de información que permita registrar y caracterizar a los vendedores informales, de tal forma que se les permitan acceder a programas de normalización de su actividad. Asimismo, el documento de la política pública de vendedores informales, en su página 29 describe: Implementación de sistemas de información o herramientas de identificación como lo es la carnetización, programas de registro y autorización para el ejercicio de la actividad, programas de acceso a la educación y a la seguridad social, de formalización laboral o acceso al empleo y de formalización empresarial, programas de reubicación y asistencia técnica. (República de Colombia – Gobierno Nacional, 2020)

También existe la ley 1988 de 02 de agosto de 2019 “POR LA CUAL SE ESTABLECEN LOS LINEAMIENTOS, PARA LA FORMULACION, IMPLEMENTACION Y EVALUACIÓN DE

UNA POLÍTICA PÚBLICA DE LOS VENDEDORES INFORMALES Y SE DICTAN OTRAS DISPOSICIONES”. Donde en su artículo 3 detalla:

ARTÍCULO 3°. Para los efectos de la presente ley, los vendedores informales se clasifican de la siguiente manera:

- a) Vendedores Informales Ambulantes: Los que realizan su labor, presentan diversas expresiones artísticas o prestan sus servicios recorriendo las vías y demás espacios de uso público, sin estacionarse temporal o permanentemente en un lugar específico, utilizando sus capacidades, un elemento móvil portátil o su propio cuerpo para transportar las mercancías;
- b) Vendedores Informales Semiestacionarios: Los que realizan su labor recorriendo las vías y demás espacios de uso público, estacionados de manera transitoria en un lugar, con la facilidad de poder desplazarse a otro sitio distinto en un mismo día, utilizando elementos, tales como carretas, carretillas, tapetes, telas, maletas, cajones rodantes o plásticos para transportar las mercancías;
- c) Vendedores Informales Estacionarios: Son las personas que para ofrecer sus bienes o servicios se establecen de manera permanente en un lugar determinado del espacio público, previamente definido para la respectiva autoridad municipal o distrital, mediante la utilización de kioscos, toldos, vitrinas, casetas o elementos similares;
- d) vendedores informales periódicos: Realizan sus actividades en días específicos de la semana o del mes, o en determinadas horas del día en jornadas que pueden llegar a ser inferiores a las ocho horas;

e) Vendedores informales ocasionales o de temporada: Realizan sus actividades en temporadas o períodos específicos del año, ligados a festividades, o eventos conmemorativos, especiales o temporadas escolares o de fin de año;

f) Temporalidad: la expresión temporal para efectos de la presente ley se refiere al término de implementación de las políticas de reubicación o formalización a iniciativa de los entes responsables, bajo ninguna circunstancia se podrá interpretar la expresión temporal como un plazo perentorio impuesto por la administración a los vendedores informales. (República de Colombia – Gobierno Nacional, 2019)

Toda esta información debe ser registrada y gestionada a través de bases de datos que almacenen información estructurada, así como información georreferenciada, entendiendo información estructurada como lo son datos alfanuméricos que se relacionan entre sí para dar sentido y orden a los datos almacenados, y la información georreferenciada que corresponde a datos que cuentan con coordenadas geográficas que permiten ubicarlos sobre la superficie de la tierra. En este aspecto se ha tomado la decisión de usar como motor de base de datos a Postgresql, el cual ofrece las siguientes características:

- a. Es open source, lo que permite reducir los costos de licenciamiento.
- b. Es multiplataforma.
- c. Es reconocida a nivel mundial por su estabilidad y escalabilidad tanto vertical como horizontal.

- d. Cumple con el protocolo ACID (acrónimo en inglés que significa Atomicity, Consistency, Isolation and Durability, y que en español traduce Atomicidad, Consistencia, Aislamiento y Durabilidad).
- e. Permiten el registro y gestión de datos georreferenciados a través de POSTGIS, extensión que agrega soporte para objetos geográficos.

Para facilitar el registro, consulta y gestión de la información estructurada y espacial que se planea manipular es necesario el desarrollo de una aplicación web responsive, que, según Wikipedia, corresponde una filosofía de diseño y desarrollo de páginas web que permite adaptar la apariencia de las mismas al dispositivo que se esté utilizando para visualizarlas.

A nivel de desarrollo de software este se va a realizar utilizando Python, lenguaje de programación interpretado y multiparadigma, que permite y soporta la programación orientada objetos, considerado muy versátil y cuyo principal enfoque es maximizar la legibilidad del código, cuyo enfoque inicial fue para procesos de análisis de datos para BIG DATA y MACHINE LEARNING, pero que ha probado ser igualmente útil para desarrollo de aplicaciones de escritorio y web a través del framework Django. Dentro de sus bondades se pueden mencionar:

- a. Lenguaje sencillo y legible. Permite implementar el paradigma de programación orientada a objetos (es multiparadigma).
- b. Es multiplataforma e interpretado.
- c. Es open source, con una amplia comunidad de desarrolladores mejorándolo constantemente.

Para lo concerniente a la captura y gestión de la información geográfica se va a utilizar el componente QGIS, cuyas ventajas son:

- a. Plataforma de código libre, bajo licencia GNU GLP.
- b. Se conecta a bases de datos Postgresql y a su componente postgis.
- c. Soporta múltiples formatos vectoriales y raster
- d. Se integra y extiende con Python a través del componente PyQgis

5. METODOLOGÍA

Para el desarrollo de la solución propuesta y explicada en la formulación del problema, se requiere el uso de una metodología ágil que permita identificar y priorizar entregables basado en el mayor beneficio que aportan, así como aprovechar al máximo los recursos humanos, técnicos y tecnológicos. Así mismo debido a las características de la problemática a atender, la cual se basa en normas y políticas públicas que están sujetas a cambios constantes, es recomendable contar una metodología que se ajuste mejor a entornos cambiantes como lo es KANBAN.

KANBAN es un concepto de producción justo-a-tiempo (JIT). KANBAN es una técnica de gestión de cambio sencilla y visual proveniente de la filosofía Lean, donde los recursos deciden cuándo y cuánto trabajo se comprometen a hacer. Los recursos toman el trabajo cuando están listos, en lugar de tener que recibirlo desde el exterior. Kanban enfatiza la respuesta al cambio por sobre seguir un plan (generalmente Kanban permite una respuesta más rápida que Scrum).

KANBAN es un método de gestión donde se desarrolla exactamente aquella cantidad de trabajo que el equipo es capaz de asumir. Es un sistema de gestión del trabajo en curso, que sirve principalmente para asegurar una producción continua, buscando evitar sobrecargas en el equipo de desarrollo.

Las características principales de KANBAN son:

1. Permite visualizar el flujo de trabajo (workflow), ya que divide el trabajo en actividades más pequeñas. Estas actividades se representan a través de tarjetas que se colocan en un

tablero. En el tablero se utilizan columnas con nombres, que permiten identificar o ilustrar en qué etapa del proceso o flujo se encuentra. A continuación, se presenta una imagen que describe de manera gráfica la estructura de un tablero KANBAN:

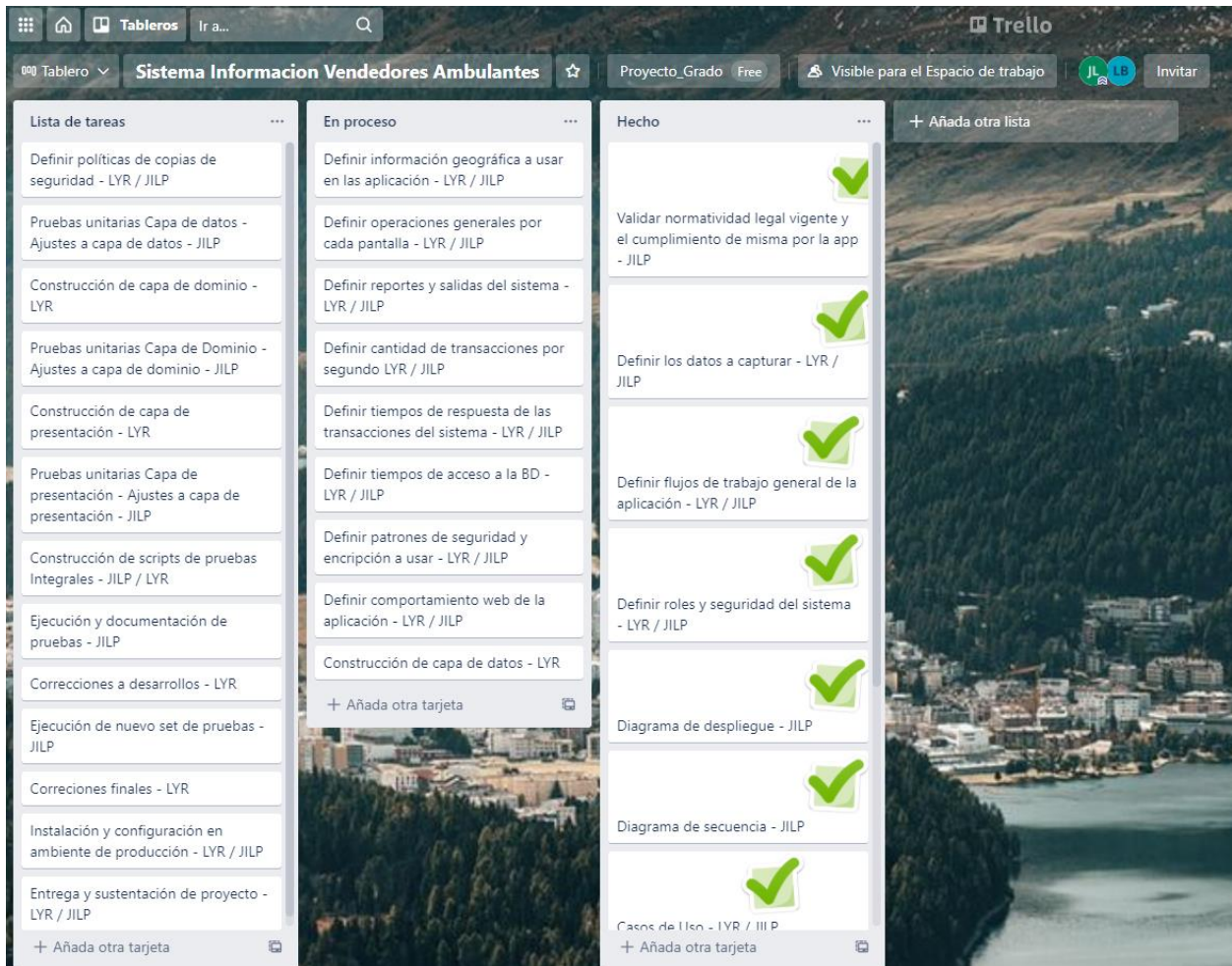


Ilustración 1 - tablero KANBAN Proyecto sistema de información para el registro y gestión de vendedores informales en los municipios de Colombia – Imagen tomada de tablero creado en herramienta TRELLO

(<https://trello.com/es>)

2. Limita el trabajo en progreso (WIP), es decir, limita la cantidad de actividades que pueden ser realizados a la vez en cada columna del flujo de trabajo (workflow)
3. Mide el tiempo de espera o “Lead Time”, que corresponde al tiempo promedio para completar cada actividad, es decir, el tiempo que toma pasar de la columna de inicio hasta la columna de final. KANBAN busca optimizar los tiempos reduciendo el tiempo de espera a la mínima expresión y de la manera más predecible posible.
4. Establecer políticas explícitas de procesos y actividades como, por ejemplo:
 - a. Quien mueve las actividades en el tablero
 - b. Cada cuanto se revisa el tablero.
 - c. Definir la periodicidad, horario y duración de las reuniones de seguimiento.
 - d. El detalle de la información que se registrará en cada tarjeta.
 - e. Definir la clasificación que se le podría dar a las actividades o tareas. Esta clasificación sirve para definir cuáles de las tareas o actividades pueden generar problemas en la ejecución del proyecto por retrasarse en su entrega.
5. Usar modelos para evaluar oportunidades de mejora permanentemente.

Aunque KANBAN no exige roles, si recomienda que se ejecuten ciertas responsabilidades para asegurar un buen funcionamiento. Estas responsabilidades se centran en los siguientes 2 roles:

- Service Request Manager. Este rol se encarga de gestionar la demanda y requisitos dentro de KANBAN. Maneja las relaciones con los stakeholders y fomenta la transparencia dentro del proyecto en torno a la priorización del trabajo. Puede semejarse al rol de product owner de SCRUM.

- Service Delivery Manager. Es responsable del flujo de trabajo dentro de KANBAN, facilitar las reuniones o KANBAN MEETINGS, el Delivery Planning, el cambio y las actividades de mejora continua. Se asemeja al rol de SCRUM MASTER.

Para el desarrollo de la aplicación propuesta en el presente documento, se va a aplicar KANBAN de la siguiente manera:

a. Conformación del equipo

Se contará con los siguientes roles:

1. Service Request Manager: Rol desempeñado por el alumno Luis Yeferson Reyes.
2. Service Delivery Manager: Rol desempeñado por el alumno Jorge Iván Lozano.
3. Equipo de desarrollo: El equipo está conformado por los estudiantes Luis Yeferson Reyes y Jorge Iván Lozano

- b. Se ha definido una (1) reunión semanal para atender cada uno de los pasos y entregables del proyecto. La siguiente figura presenta el cronograma creado para apoyar la gestión de todas las actividades del proyecto:

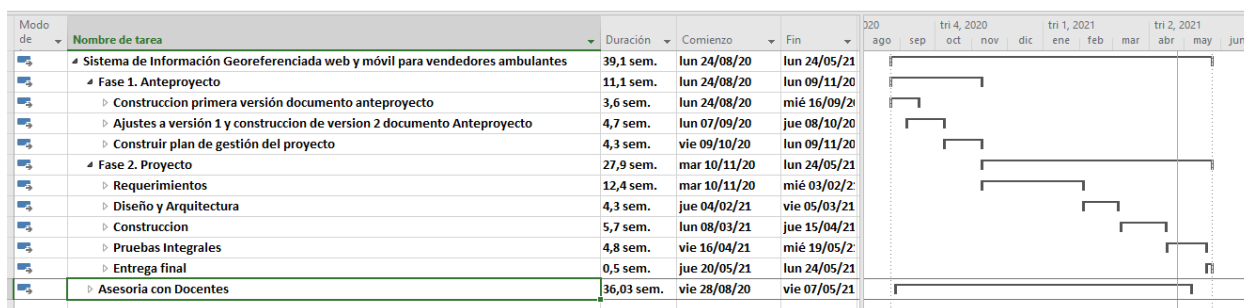


Ilustración 2 - Cronograma para el desarrollo del sistema de información para el registro y gestión de vendedores informales en los municipios de Colombia – Fuente propia

- c. Se levantarán los requerimientos funcionales usando Casos de Uso.
- d. Se definirá y hará una revisión permanente al tablero KANBAN para la priorización de entregables.
- e. Se implementará y gestionará un taskboard para llevar un control gráfico de las tareas que están por iniciar, en ejecución y terminadas.

6. PROCESO DE SOFTWARE

6.1 Requerimientos Funcionales

En esta sección se describen con más detalle las funcionalidades que debe tener el sistema:

Número de requisito	RF-01		
Nombre de requisito	Registrar un Vendedor Informal		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Alcaldía.		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
Descripción	La aplicación debe permitir el registro de vendedores informales. Debe incluir información básica de cliente, la ubicación del sitio donde realizan la actividad de venta informal y en que consiste la actividad que realizan. El registro debe poder ser realizado por los encuestadores y administradores de la aplicación.		

Tabla 1 - Requerimiento funciona Registro de Vendedores – Fuente propia

Número de requisito	RF-02		
Nombre de requisito	Actualizar / Eliminar un Vendedor Informal		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Alcaldía.		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
Descripción	La aplicación debe permitir la actualización y eliminación de información relacionada con vendedores informales. Dichas operaciones se deben poder hacer sobre la información básica de cliente, la ubicación del sitio donde realizan la actividad de venta informal y en que consiste la actividad que realizan. Estas		

	operaciones solo pueden ser realizadas por los administradores de la aplicación.
--	--

Tabla 2 - Requerimiento funcional Actualizar / Eliminar Vendedor Informal - Fuente propia

Número de requisito	RF-03		
Nombre de requisito	Generar Reportes		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Alcaldía.		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
Descripción	La aplicación debe permitir la consulta de información relacionada con los vendedores informales, por su documento de identidad, ubicación donde realizan la actividad laboral, etnia, nivel de educación, genero, tipo de documento y documento de identidad. Adicionalmente el resultado de la consulta debe poder ser exportado a un archivo en formato PDF o EXCEL. Estas operaciones solo pueden ser realizadas usuarios con perfil de administradores de la aplicación, y aquellos que hayan sido autorizados para generar reportes.		

Tabla 3 - Requerimiento funcional Generar Reportes - Fuente propia

6.2 Requerimientos No Funcionales

A continuación, se describen los requerimientos no funcionalidades que debe tener en cuenta el sistema:

ESCENARIO DE CALIDAD #	1	STAKEHOLDER: Encuestador	PRIORIDAD: Alta
Atributo de Calidad	Seguridad		
Justificación	El ingreso de información a la aplicación a través de la web, requiere un nivel de seguridad que garantice el cumplimiento de autorización y autenticación para el stakeholder		
Fuente	Usuario Final		

Estímulo	El usuario ingresa a la aplicación para adicionar información de vendedores informales del municipio o región.
Artefacto	Aplicación
Ambiente	La aplicación se encuentra en operación normal
Respuesta	Se utilizan mecanismos de verificación de identidad, a través de validación de usuario y contraseña, dejando ingresar al usuario solo en caso de que haya sido correcto el proceso de autenticación.
Medida de la Respuesta	El 100% de las veces que un usuario ingresa información de un vendedor informal se validó su identidad y se verificó si estaba autorizado para realizar dicha operación.

Tabla 4 - Requerimiento no funcional Seguridad usuario Encuestador - Fuente propia

ESCENARIO DE CALIDAD #	2	STAKEHOLDER: Administrador	PRIORIDAD: Alta
Atributo de Calidad	Seguridad		
Justificación	El ingreso, modificación o eliminación de información a la aplicación a través de la web, requiere un nivel de seguridad que garantice el cumplimiento de autorización y autenticación para el stakeholder		
Fuente	Usuario Final		
Estímulo	El usuario ingresa a la aplicación para adicionar, modificar o eliminar información de vendedores informales del municipio o región.		
Artefacto	Aplicación		
Ambiente	La aplicación se encuentra en operación normal		
Respuesta	Se utilizan mecanismos de verificación de identidad, a través de validación de usuario y contraseña, dejando ingresar al usuario solo en caso de que haya sido correcto el proceso de autenticación.		
Medida de la Respuesta	El 100% de las veces que un usuario ingresa, modifica o elimina información de un vendedor informal se validó su identidad y se verificó si estaba autorizado para realizar dicha operación.		

Tabla 5 - Requerimiento no funcional Seguridad usuario Administrador - Fuente propia

ESCENARIO DE CALIDAD #	3	STAKEHOLDER: Vendedor / Alcaldía / Departamento	PRIORIDAD: Alta
Atributo de Calidad	Seguridad		
Justificación	El ingreso de información a la aplicación a través de la web, requiere un nivel de seguridad que garantice el cumplimiento de autorización y autenticación para el stakeholder		
Fuente	Usuario Final		
Estímulo	El usuario ingresa a la aplicación para consultar información de vendedores informales del municipio o región.		
Artefacto	Aplicación		
Ambiente	La aplicación se encuentra en operación normal		
Respuesta	Se utilizan mecanismos de verificación de identidad, a través de validación de usuario y contraseña, dejando ingresar al usuario solo en caso de que haya sido correcto el proceso de autenticación.		
Medida de la Respuesta	El 100% de las veces que un usuario consulta información de un vendedor informal se validó su identidad y se verificó si estaba autorizado para realizar dicha operación.		

Tabla 6 - Requerimiento no funcional Seguridad otros usuarios - Fuente propia

ESCENARIO DE CALIDAD #	4	STAKEHOLDER: Encuestadores / Administrador / Vendedor / Alcaldía / Departamento	PRIORIDAD: Alta
Atributo de Calidad	Disponibilidad		
Justificación	La razón de ser de la aplicación es poder acceder en cualquier momento a registrar, modificar, eliminar y/o consultar información registrada en el sistema.		

Fuente	Encuestadores / Administrador / Vendedor / Alcaldía / Departamento
Estímulo	Un interesado intenta acceder a la aplicación
Artefacto	Aplicación
Ambiente	La aplicación de encuentra en operación normal
Respuesta	Al momento de la comunicación la aplicación se encuentra en línea y es capaz de permitir actividades de ingreso, modificación, eliminación o consulta de información (según el perfil y permisos definidos para cada usuario).
Medida de la Respuesta	Como mínimo el 99,99% de las veces en que solicita acceso a la aplicación, está disponible para atenderla oportunamente.

Tabla 7 - Requerimiento no funcional Disponibilidad - Fuente propia

ESCENARIO DE CALIDAD #	5	STAKEHOLDER: Encuestadores / Administrador / Vendedor / Alcaldía / Departamento	PRIORIDAD: Media
Atributo de Calidad	Interoperabilidad		
Justificación	La aplicación debe poder consultarse en las últimas versiones de Chrome, Mozilla Firefox, Microsoft Edge y Safari		
Fuente	Usuario Final		
Estímulo	El usuario ingresa a la aplicación web a través de su computador o móvil, para registrar, modificar, eliminar o consultar información, sobre los vendedores informales.		
Artefacto	Aplicación		
Ambiente	La aplicación se encuentra en operación normal.		
Respuesta	El usuario hace uso de la aplicación para registrar, modificar, eliminar y consultar información de los vendedores informales (según el perfil y permisos definidos para cada usuario).		

Medida de la Respuesta	El 100% de las acciones de registro, modificación, eliminación y consulta de los navegadores y dispositivos móviles se realiza de manera exitosa.
-------------------------------	---

Tabla 8 - Requerimiento no funcional Interoperabilidad - Fuente propia

ESCENARIO DE CALIDAD #	6	STAKEHOLDER: Encuestadores / Administrador / Vendedor / Alcaldía / Departamento	PRIORIDAD: Media
Atributo de Calidad	Usabilidad		
Justificación	La aplicación debe permitir una interacción fácil e intuitiva, a través de un diseño web enfocado a usuarios con pocos conocimientos en tecnología.		
Fuente	Usuario Final		
Estímulo	El usuario puede ingresar a la aplicación de información a través de la web para registrar, modificar, actualizar y/o consultar información (según el perfil y permisos definidos para cada usuario).		
Artefacto	Aplicación		
Ambiente	La aplicación se encuentra en operación normal.		
Respuesta	El usuario es capaz de realizar la tarea que se propuso de manera correcta.		
Medida de la Respuesta	La tasa de errores en pruebas realizadas con usuarios finales novatos presentada es menor al 25%, esta se mide relacionando el número de intentos fallidos sobre el número de intentos totales. Cabe aclarar que error se refiere a equivocaciones del usuario.		

Tabla 9 - Requerimiento no funcional Usabilidad - Fuente propia

ESCENARIO DE CALIDAD #	7	STAKEHOLDER	PRIORIDAD:
-------------------------------	----------	--------------------	-------------------

Atributo de Calidad	Escalabilidad
Justificación	Se espera un crecimiento del 15% anual en el registro y gestión de la información de vendedores informales, lo que genera la necesidad de incluir en la estructura misma de la aplicación la posibilidad de soportar un número creciente de usuarios accediendo a recursos concurrentemente.
Fuente	Equipo de proyecto
Estímulo	Múltiples entes territoriales registran, modifican, eliminan y/o consultan información en la aplicación.
Artefacto	Aplicación
Ambiente	La aplicación se encuentra en estado de estrés.
Respuesta	Se procesan todas las solicitudes y se realizan las acciones correspondientes de acuerdo al tipo de acción ejecutada.
Medida de la Respuesta	La aplicación debe procesar en promedio 50 eventos concurrentes en intervalos de 10 segundos.

Tabla 10 - Requerimiento no funcional Escalabilidad - Fuente propia

6.3 Diseño y Arquitectura

Con el objetivo de implementar una arquitectura que, de respuesta a los requerimientos funcionales y no funcionales identificados para la problemática relacionada con los vendedores informales, que facilite también el responder a contar con componentes de innovación, pero que adicionalmente facilite la construcción y posteriormente mantenimiento de la solución se ha propone el uso de una arquitectura basado en el patrón Modelo – Vista – Controlador (Model-View-Controller). A continuación, se muestra la representación gráfica de un patrón MVC:

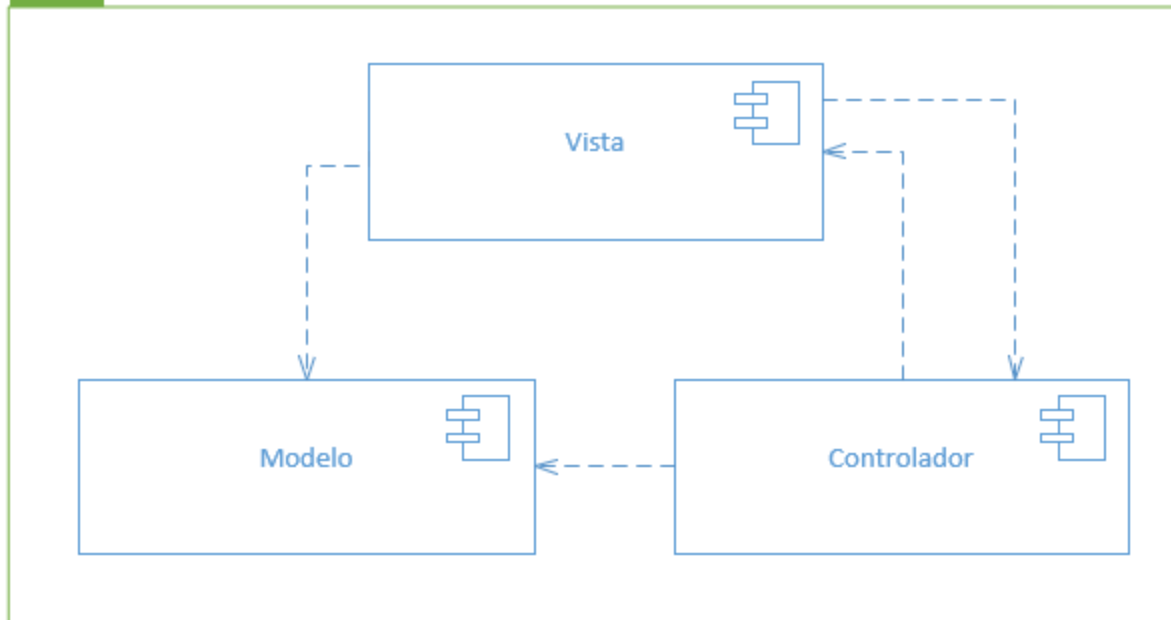


Ilustración 3- Modelo-Vista-Controlador - Fuente propia

Este patrón permite separar responsabilidades de la presentación y lógica de negocio obteniendo un código más entendible y mantenible. En este patrón los cambios realizados por el usuario en las páginas (View) se notifican al controlador (Controller) quien procesa el evento y actualiza el modelo (Model). El controlador es quien accede a la capa de negocio mediante los servicios que ésta ofrece.

En otras palabras, esta arquitectura define con claridad la cantidad mínima de capas que debe tener una aplicación, propone y soporta el uso del concepto de capas cerradas, donde para ir a una capa inferior debe hacerse solo a través de su capa inmediatamente anterior, permite implementar separación de responsabilidades que a su vez facilita el desarrollo, mantenimiento y pruebas de

cada capa, así como lograr implementar un alto nivel de seguridad al admitir tener cada capa en servidores independientes y aislados.

Para soportar la persistencia de la solución planteada, adicional que ofrezca la posibilidad de almacenar información geográfica, se va a utilizar un motor de bases de datos relacional como lo es Postgresql.

En cuanto a las herramientas para el desarrollo de la solución se seleccionó a PYTHON, lenguaje que ofrece una sintaxis de fácil y rápido entendimiento y, que se está convirtiendo en un referente no solo para el desarrollo de aplicaciones desktop, de big data y de machine learning, sino también para el desarrollo WEB con el uso de frameworks. Al respecto, se indagaron inicialmente varios frameworks compatibles con PYTHON y que soportara el patrón Modelo-Vista-Controlador. Estos fueron: DJANGO, FLASK y PYRAMID, seleccionándose DJANGO como el framework más adecuado para el objetivo propuesto.

Durante el proceso de aprendizaje del uso de PYTHON y DJANGO, se presentaron dificultades en el entendimiento y uso del framework, también por el poco conocimiento inicial del lenguaje, lo que estaba demorando el desarrollo de la solución. Ante lo anterior se buscaron alternativas que permitieran continuar con el uso del lenguaje seleccionado y con la aplicación del patrón arquitectural seleccionado. En el proceso de investigación se logró encontrar una herramienta conocida como ODOO (anteriormente OpenERP). ODOO es una herramienta ERP y CRM WEB integrada de código abierto, que cuenta con un framework para la ampliación de funcionalidades de la aplicación, pero que a la vez permite la creación de aplicaciones independientes, para la web y desarrolladas en PYTHON, brindando herramientas y componentes que agilizan el desarrollo.

6.4 Diagrama de despliegue

En la ilustración que se presenta a continuación, se detallan las tecnologías usadas para la construcción de la solución, y que siguen las recomendaciones dadas por la arquitectura.

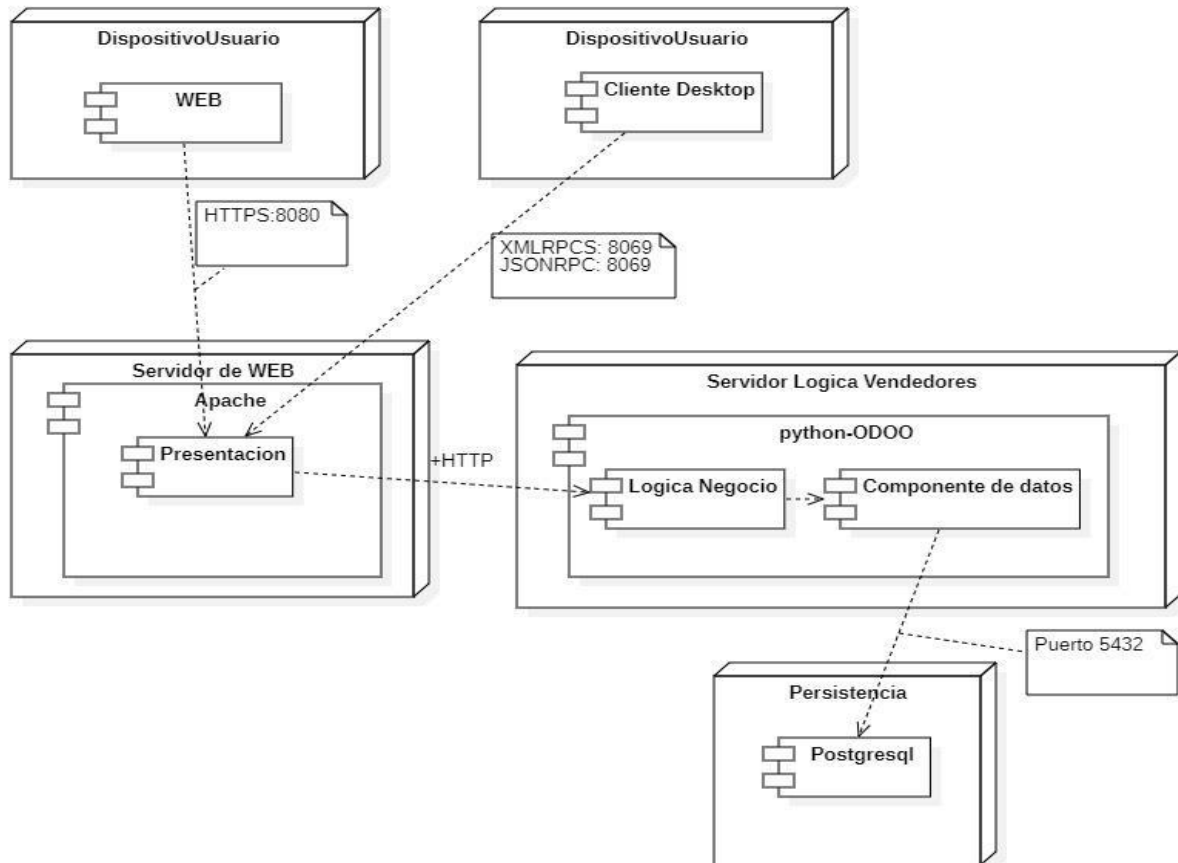


Ilustración 4 - Diagrama de Despliegue - Fuente propia

La aplicación web está construida utilizando ODOO. ODOO es una solución integral de ERP y CRM que cuenta con un framework de desarrollo que utiliza una arquitectura de tres capas MVC (Modelo-Vista-Controlador). La capa lógica maneja el acceso a datos y el procesamiento. Es capaz

de agregar datos con operaciones de consulta de entidades, creación, modificación y eliminación. La capa de almacenamiento encargada de la persistencia de la información utiliza postgresql como motor de base de datos

La decisión de utilizar ODOO como ambiente de desarrollo se basa en la necesidad de contar con una herramienta que facilitara el proceso de desarrollo y que permitiera atender las características particulares del proyecto y los requerimientos del mismo. Este ambiente corre sobre tecnologías Python, HTML, XML y Javascript y utiliza patrones de diseño ya probados y alineados con el diseño realizado de la aplicación en tres capas.

Por otra parte, el tiempo de desarrollo de la solución es limitado razón por la cual utilizar una herramienta que simplifique el desarrollo es crítico; como se mencionó, ODOO genera la mayoría de código permitiendo al desarrollador centrarse en la lógica de negocio y en la persistencia de los datos.

6.5 Caso de Uso Arquitecturalmente Relevante

Los casos de uso definen el comportamiento de la aplicación en términos de las interacciones del usuario. Los casos de uso identificados representan las principales acciones o interacciones del usuario en relación al registro y mantenimiento de la información relacionada con los vendedores informales, la asignación de beneficios y la generación de reportes. A continuación, se presenta la ilustración que explica el caso de uso arquitecturalmente relevante:

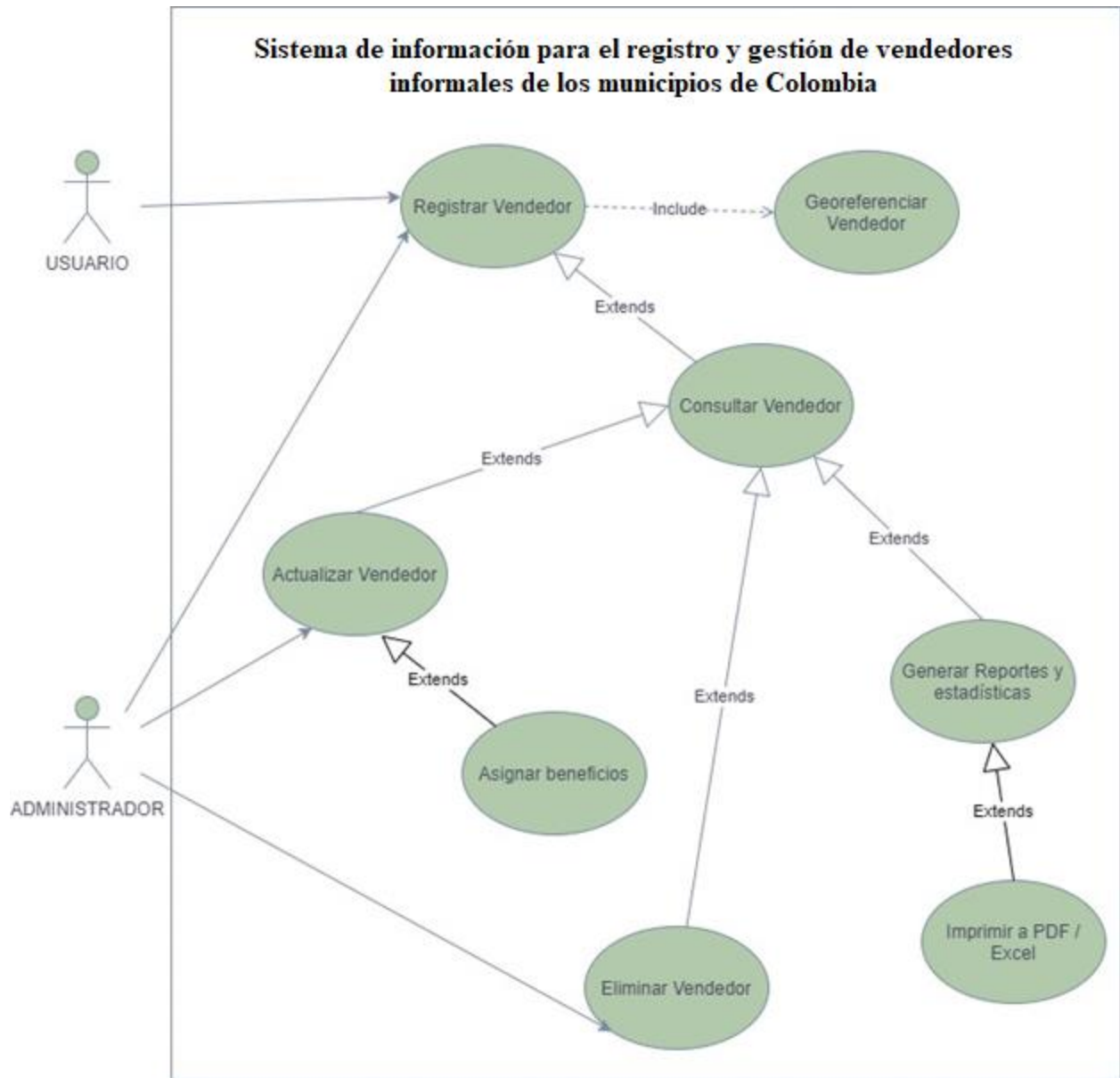


Ilustración 5 - Caso de Uso Arquitecturalmente Relevante - Fuente propia

A continuación, se detallan los casos de uso expuestos en el diagrama de casos de uso arquitecturalmente relevante:

Título del Caso de Uso	Registrar Vendedor	ID del Caso de Uso	CU_01
Descripción General del Caso de Uso			

Permite registrar en la aplicación de los vendedores informales.	
Entidades Involucradas	
Vendedor / Municipio / Género / Grupo Étnico / Educación / Actividad	
Precondiciones	
Ingresar a la Aplicación	
Flujo normal de Eventos	
1	Seleccionar la opción “Registro de vendedor”
2	Se carga el formulario para el registro de la información de los vendedores informales
3	Se diligencian los campos solicitados
4	Se selecciona la opción “Guardar”
Postcondiciones principales del caso de uso	
La información del proveedor queda almacenada en la aplicación	

Tabla 11 - Caso de uso Registrar Vendedor - Fuente propia

Título del Caso de Uso	Consultar Vendedor	ID del Caso de Uso	CU_02
Descripción General del Caso de Uso			
Permite consultar en la aplicación la información de los vendedores informales.			
Entidades Involucradas			
Vendedor / Municipio / Género / Grupo Étnico / Educación / Actividad			
Precondiciones			
Ingresar a la Aplicación			
Flujo normal de Eventos			
1	En la pantalla principal, seleccionar la opción FILTROS (lista desplegable)		
2	Seleccionar el campo de interés e ingresar el valor de búsqueda. Se pueden seleccionar varios campos para mejorar la búsqueda		
3	Hacer click sobre el icono que representa una LUPA		
4	Se genera una lista con el resultado de la búsqueda		
5	Se puede seleccionar de la lista un vendedor para ver el detalle de los datos del mismo		
Postcondiciones principales del caso de uso			
Se muestra la lista de vendedores que coinciden con los datos de búsqueda			

Tabla 12 - Caso de uso Consultar Vendedor - Fuente propia

Título del Caso de Uso	Actualizar Vendedor	ID del Caso de Uso	CU_03
Descripción General del Caso de Uso			
Permite actualizar la información de un vendedor en la aplicación.			
Entidades Involucradas			

Vendedor / Municipio / Género / Grupo Étnico / Educación / Actividad	
Precondiciones	
Ingresar a la aplicación Ejecutar el caso de uso CU_02 Consultar Vendedor	
Flujo normal de Eventos	
1	Se selecciona de la lista el vendedor de interés para ver el detalle de los datos del mismo
2	Se despliega formulario para actualización de datos
3	Se realiza la actualización de datos de interés
4	Se presiona la opción “Guardar”
Postcondiciones principales del caso de uso	
La información del vendedor queda actualizada en la aplicación	

Tabla 13 - Caso de uso Actualizar Vendedor - Fuente propia

Título del Caso de Uso	Eliminar Vendedor	ID del Caso de Uso	CU_04
Descripción General del Caso de Uso			
Permite eliminar la información de un vendedor en la aplicación. Ejecutar el caso de uso CU_02 Consultar Vendedor			
Entidades Involucradas			
Vendedor / Municipio / Género / Grupo Étnico / Educación / Actividad			
Precondiciones			
Ingresar a la aplicación Ejecutar el caso de uso CU_02 Consultar Vendedor			
Flujo normal de Eventos			
1	Se selecciona de la lista el vendedor de interés para ver el detalle de los datos del mismo		
2	Seleccionar la opción ‘Suprimir’ para el vendedor que se quiere eliminar		
3	Se muestra un mensaje de confirmación		
4	Seleccionar la opción ‘Aceptar’		
Postcondiciones principales del caso de uso			
La información del vendedor es eliminada de la aplicación			

Tabla 14 - Caso de uso Eliminar Vendedor - Fuente propia

Título del Caso de Uso	Generar Reportes	ID del Caso de Uso	CU_06
Descripción General del Caso de Uso			
Permite generar reportes con la información de los vendedores registrados en la aplicación.			

Entidades Involucradas	
Vendedor / Municipio / Género / Grupo Étnico / Educación / Actividad	
Precondiciones	
Ingresar a la aplicación	
Flujo normal de Eventos	
1	Seleccionar la opción “Reportes”
2	Se selecciona el reporte de interés y se selecciona la opción “Ejecutar”
3	Se genera el reporte seleccionado
4	Se selecciona entre “Exportar a PDF” o “Exportar a Excel”
5	Se genera el archivo seleccionado
Postcondiciones principales del caso de uso	
Archivo en formato PDF o en formato Excel generado.	

Tabla 15 - Caso de uso Generar Reportes - Fuente propia

6.6 Diagrama de Secuencia

A continuación, se presentan los diferentes diagramas de secuencia, que muestran la forma como el usuario interactúa con los principales componentes de la aplicación, así como el comportamiento a nivel de componentes para los casos de uso relacionados en el ítem anterior. Las ilustraciones que se presentan a continuación, exponen los diagramas de secuencia de las funcionalidades de la aplicación:

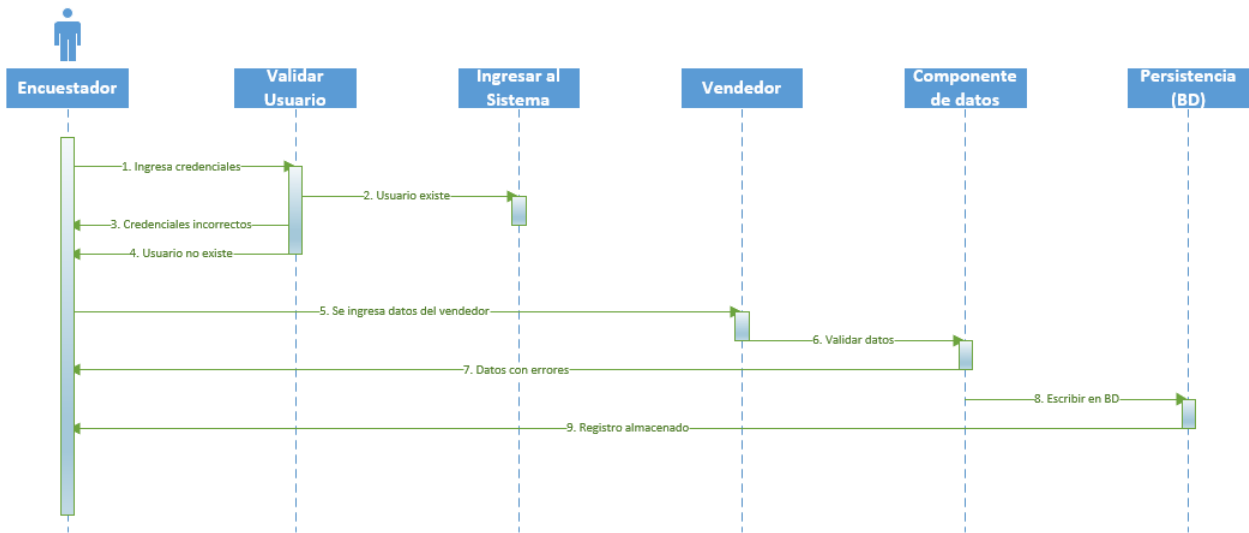


Ilustración 6 - Diagrama de secuencia Registrar Vendedor - Fuente propia

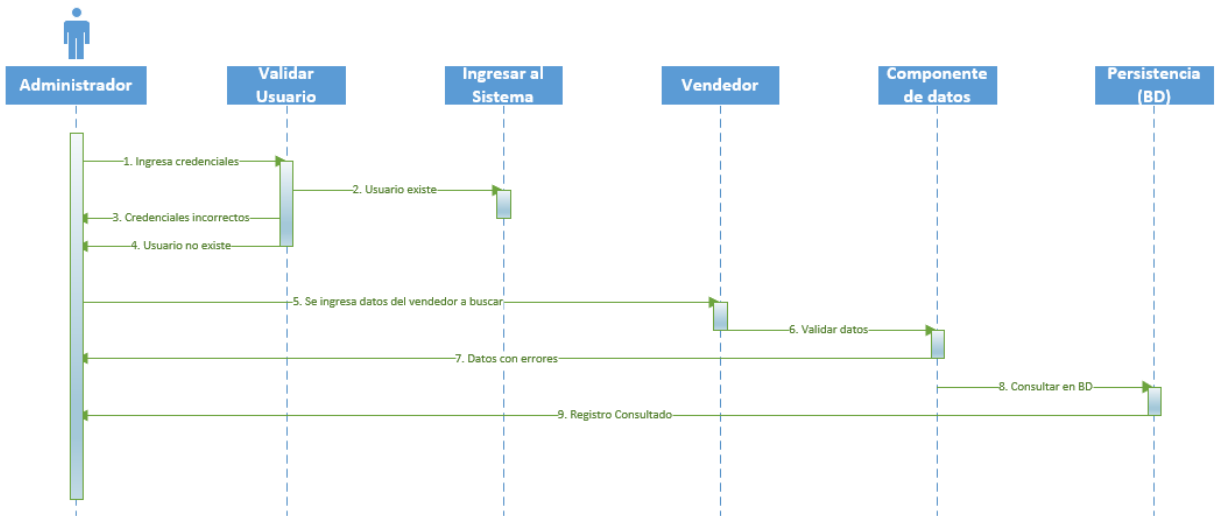


Ilustración 7 - Diagrama de secuencia Consultar Vendedor - Fuente propia

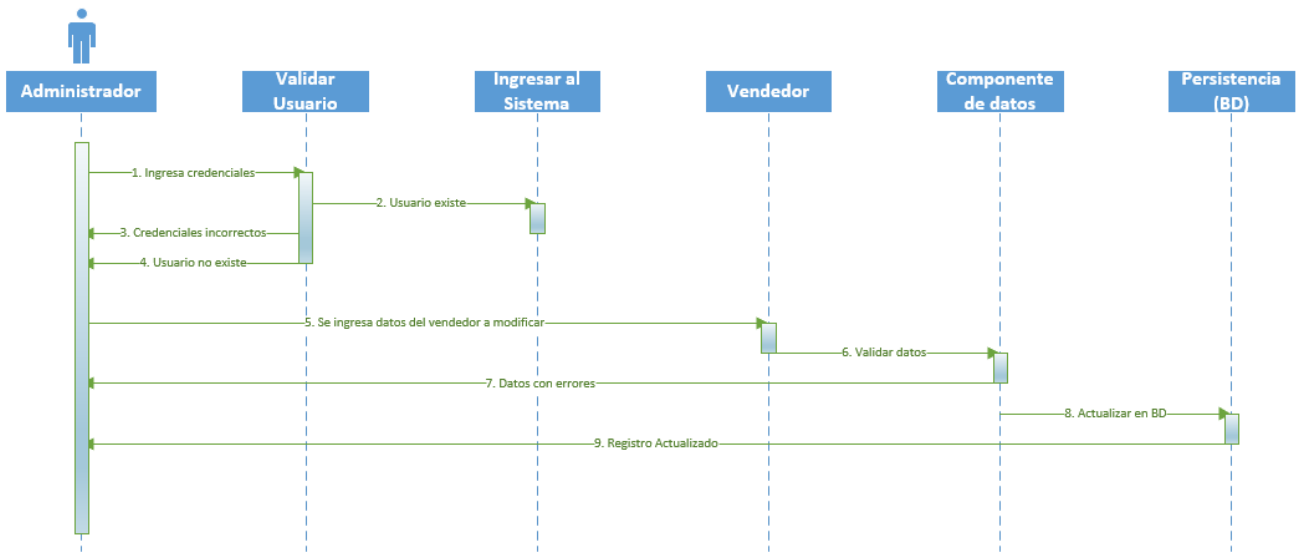


Ilustración 8 - Diagrama de secuencia Actualizar Vendedor - Fuente propia

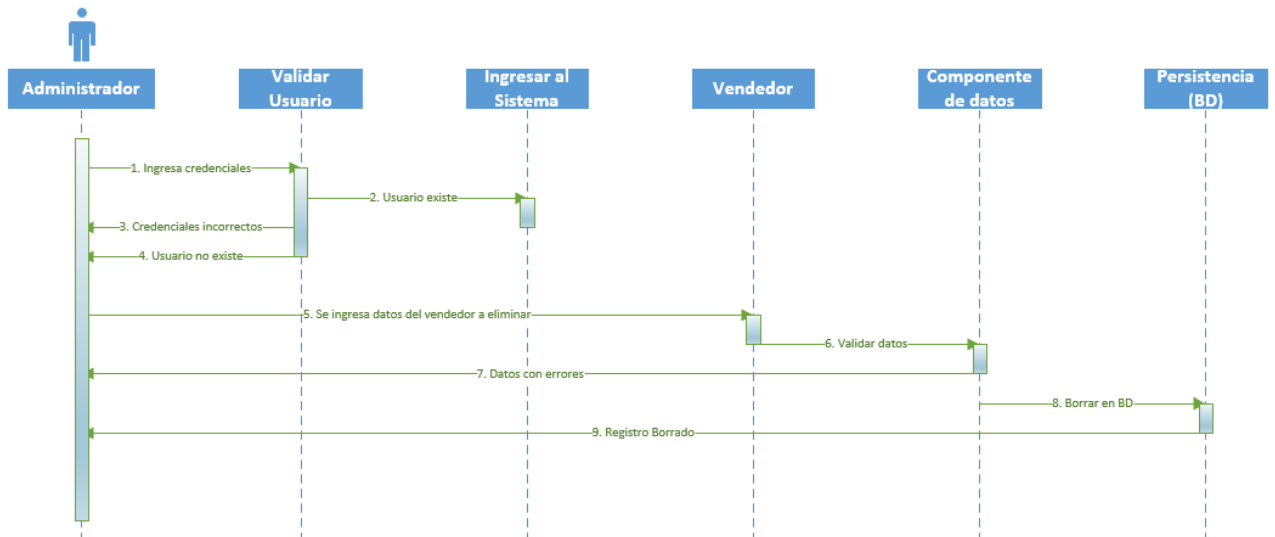


Ilustración 9 - Diagrama de secuencia Eliminar Vendedor - Fuente propia

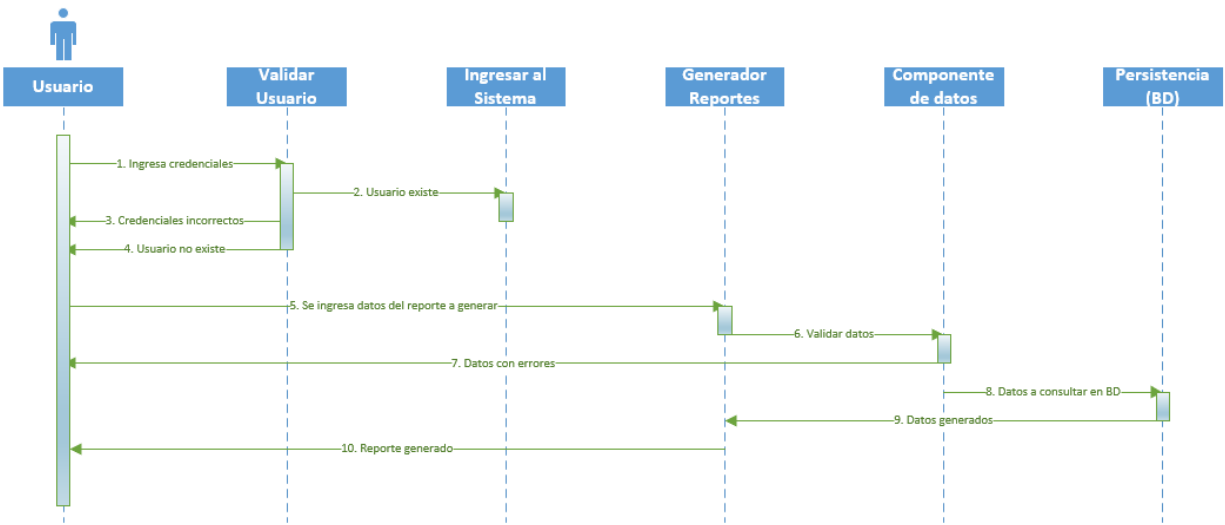


Ilustración 10 - Diagrama de secuencia Generar Reportes - Fuente propia

6.7 Diagrama de Clases

A continuación, se presenta el diagrama de objetos que describe los principales objetos que interactúan para describir el comportamiento de la solución diseñada:

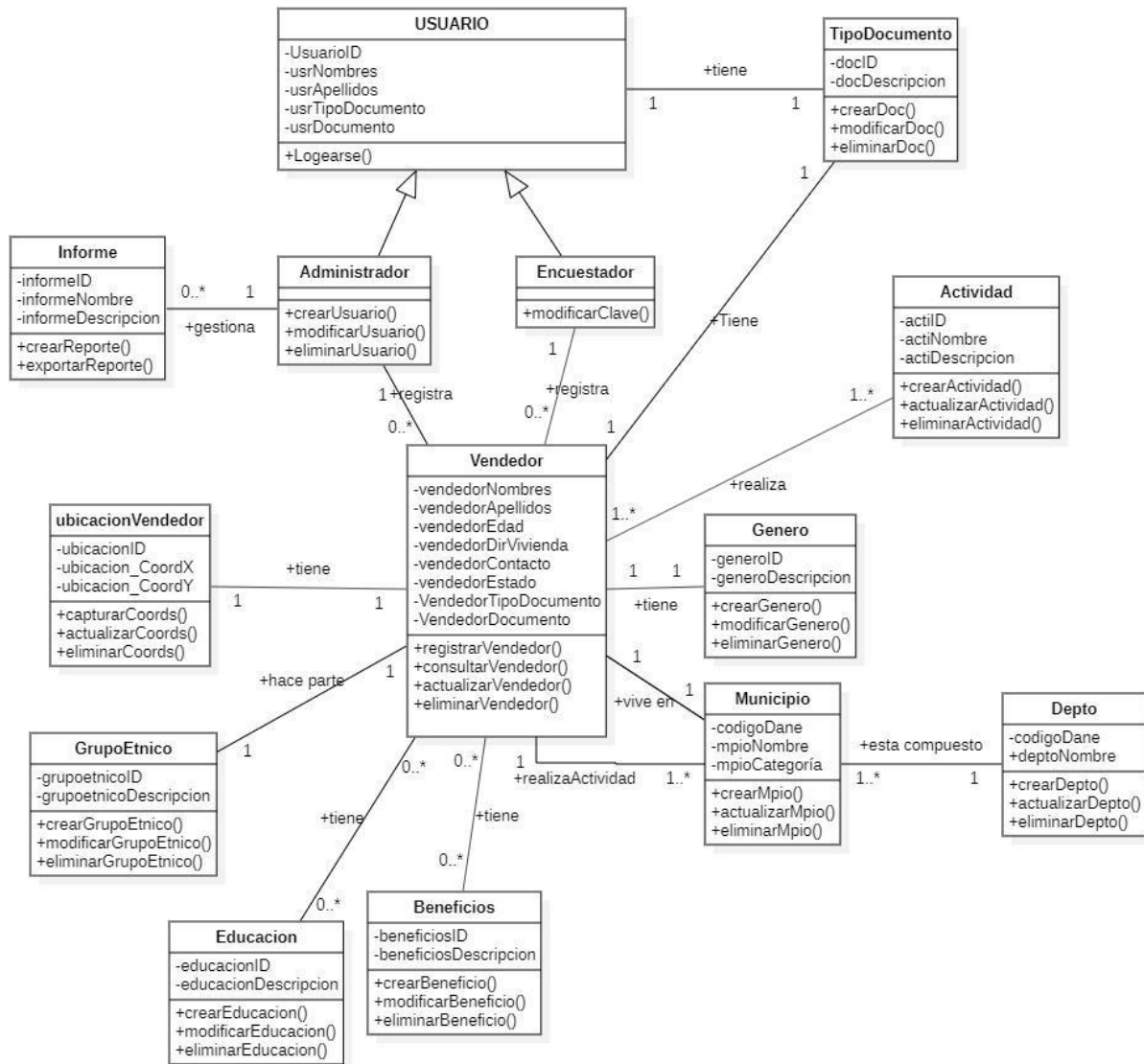


Ilustración 11 - Diagrama de Clases - Fuente propia

6.8 Arquitectura de Alto Nivel

A continuación, se presenta el diagrama que describe los principales componentes que interactúan para describir la arquitectura de Alto Nivel de la solución diseñada

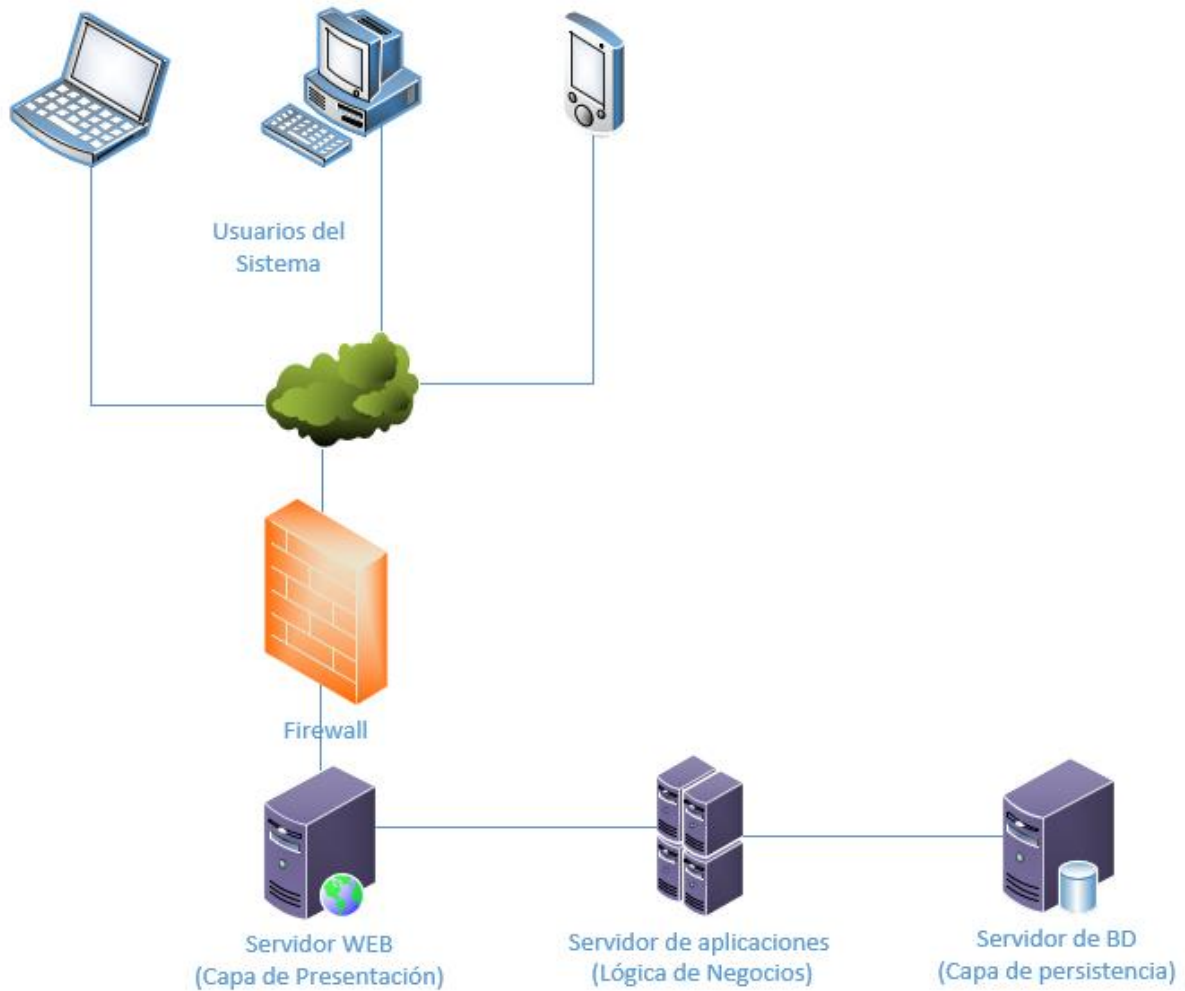


Ilustración 12 - Diagrama de Arquitectura de Alto Nivel - Fuente propia

7. CONSTRUCCIÓN

Para la materialización de la propuesta de solución, esta se construyó siguiendo una arquitectura Modelo-Vista-Controlador o MVC.

La selección de MVC, se debió principalmente a que es una arquitectura muy usada actualmente para el desarrollo de aplicaciones WEB, sobre la cual ya se tenía un conocimiento previo por parte del equipo de desarrollo y que ofrece ciertas ventajas para la construcción de la aplicación de información propuesta, cuya complejidad no es alta.

Adicionalmente, para reducir el tiempo de desarrollo, se buscaron herramientas que facilitaran la construcción de la solución, implementando de una vez el modelo arquitectural propuesto. Por lo anterior se seleccionó a ODOO (antes conocido como OpenERP), solución que ofrece un entorno de desarrollo integrado y que permite la creación de aplicaciones propias, que se pueden vincular, si así se desea, al entorno de ejecución de ODOO, y que por defecto proponen la implementación de MVC.

Para el desarrollo se usó Visual Studio Code como editor de código. Sobre este entorno de desarrollo, se instalan las extensiones:

- python
- odoo-snippets

A continuación, se muestran las ilustraciones que describen el entorno de desarrollo usado para la construcción de la aplicación:

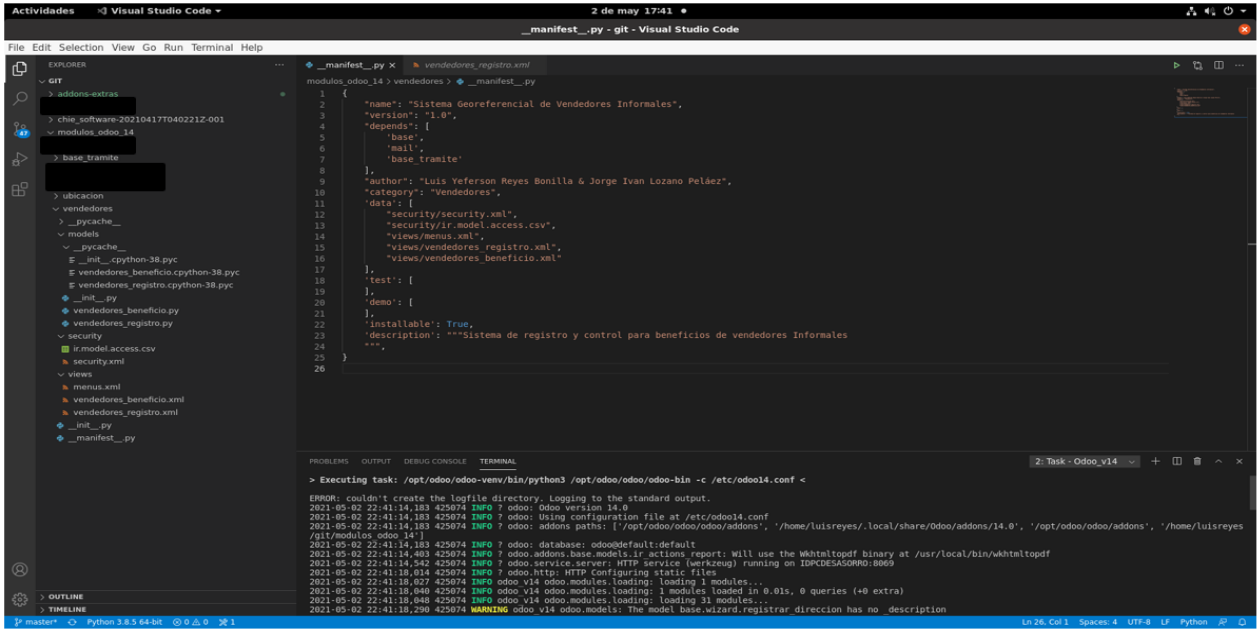


Ilustración 13 - Entorno de Desarrollo Visual Studio Code – Fuente Propia

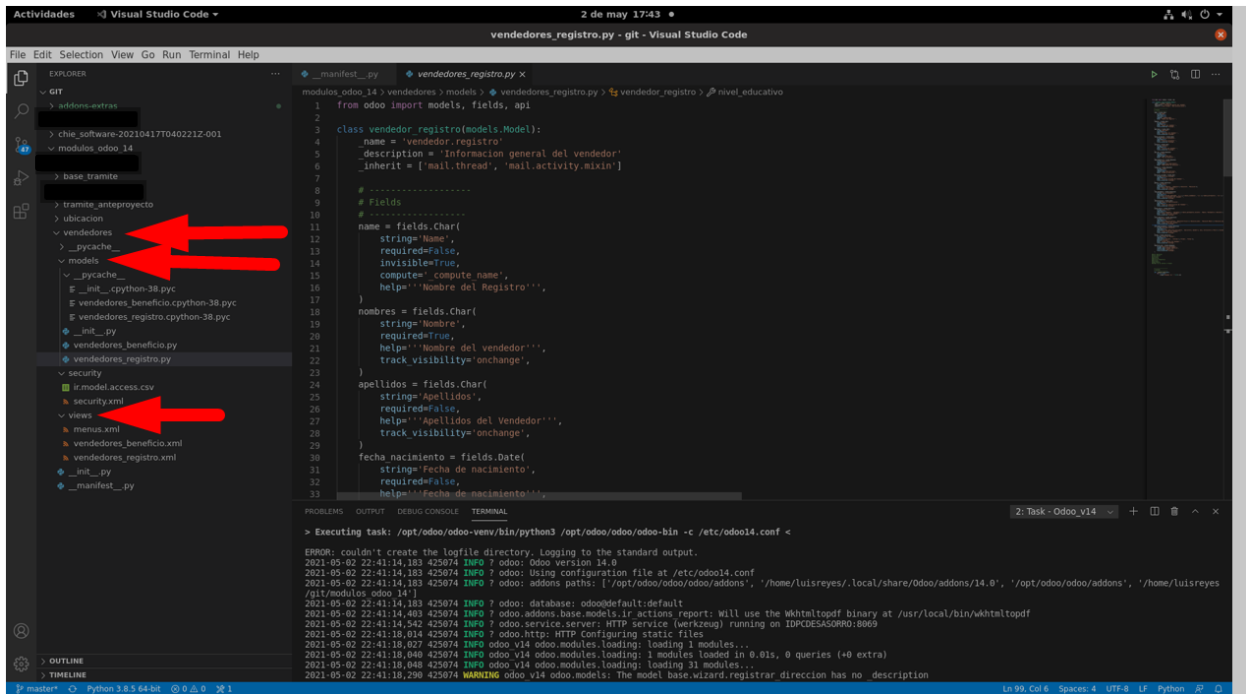


Ilustración 14 - Entorno de desarrollo, uso de MVC – Fuente Propia

Lo anterior permite validar como para extender a ODOO, así como para la construcción de aplicaciones usando las herramientas de desarrollo de ODOO apoyándose en Python como lenguaje de programación y la orientación a MVC que propone dicho lenguaje, para desarrollos WEB.

El resultado de aprovechar las bondades que ofrece el uso de los componentes de desarrollo de OdoO, se evidencian en el rápido desarrollo de soluciones como se pueden ver en las siguientes imágenes:

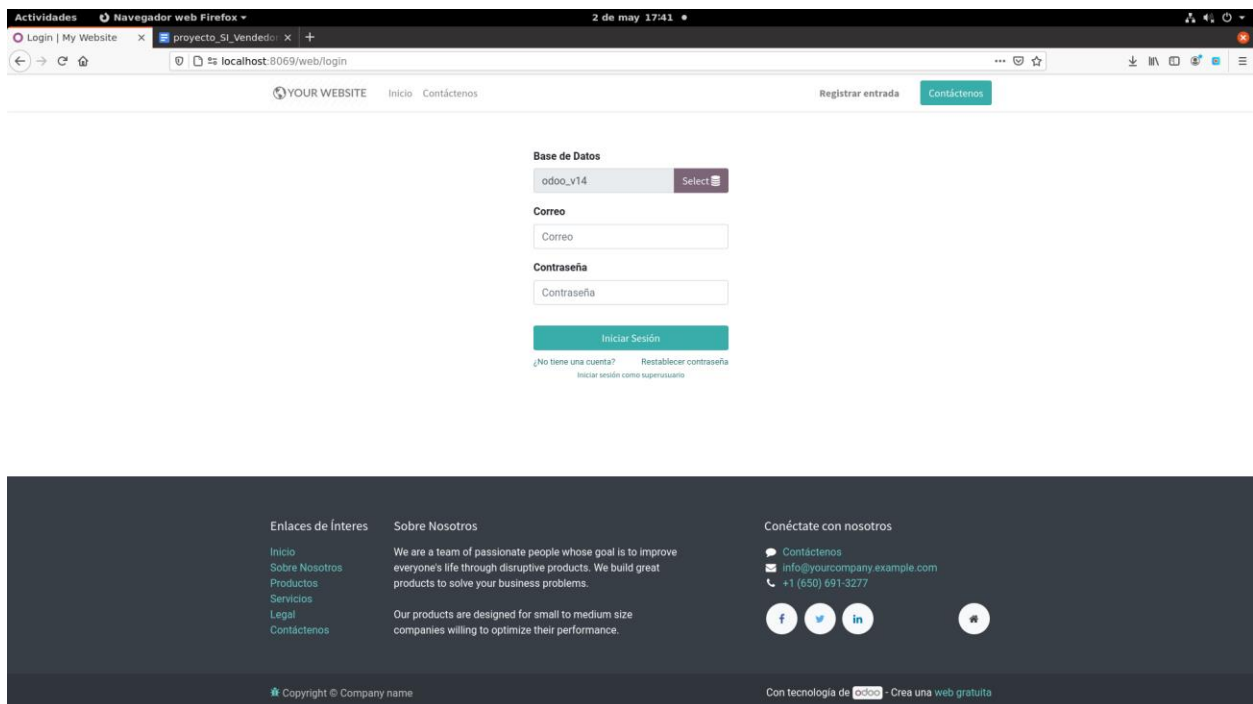


Ilustración 15 - Pantalla de Ingreso S.I. Vendedores Informales - Fuente propia.

Actividades | Navegador web Firefox | 2 de may 17:43

Nuevo - Odoo | proyecto_SI_Vendedor | localhost:8069/web#id=&action=219&model=vendedor.registro&view_type=form&cids=&menu_id=197

Sistema de Registro de Vendedores Informales | Registro de Vendedores Informales | Administrator (odoo_v14)

Registro de Vendedor / Nuevo

Guardar Descartar

Informal Formal

Nombre

Apellidos

Fecha de nacimiento

Contacto

Pais

Departamento

Ciudad

Grupo étnico

Nivel Educativo

Actividad Económica

Dirección Vivienda

Registrar Dirección

Genero

Tipo de documento

Numero de Identificación

Registrado por

Fecha de registro

Beneficios Entregados

Beneficios

Beneficio	Descripción del beneficio	Fecha de B...	Beneficio entregado por
Agregar línea			

Ilustración 16 - Formulario de registro de un vendedor informal - Fuente propia

8. PRUEBAS

A continuación, se describen y documentan las pruebas realizadas que permiten verificar que la aplicación satisface los requerimientos o cubre las necesidades para las cuales fue concebido, así como facilitar la detección de defectos y la corrección de los mismos. Adicionalmente para dar cumplimiento al objetivo de los atributos de calidad definidos para el presente proyecto, se presentan los escenarios de calidad y las actividades realizadas para tal fin.

8.1 Escenario de calidad adecuación funcional

A continuación, se describen los casos de prueba realizados sobre el sistema de información, que sustentan o soportan el cumplimiento al atributo de calidad de Adecuación Funcional

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
CASO DE PRUEBA No.	<i>CP_01</i>	VERSIÓN DE EJECUCIÓN	<i>V1</i>
CASO DE USO:	<i>CU_01</i>	MÓDULO DEL SISTEMA	Gestión de Vendedores
Objetivo del caso de prueba:	Validar que la creación o registro de un nuevo vendedor se haga correctamente en la aplicación.		
1. CASO DE PRUEBA			
a. Precondiciones			
<ul style="list-style-type: none"> • Contar con un usuario con perfil encuestador ya creado en la aplicación. • No debe existir información previa sobre el vendedor que se va a registrar. 			
b. Pasos de la prueba			
1	Seleccionar la opción “Registro de vendedores informales – Visor geográfico”		
2	Se despliega el mapa del municipio sobre el que se está haciendo la encuesta y se hace click sobre el sitio del mapa donde se ubica el vendedor		
3	Se despliega el formulario		

4	Se diligencian los datos solicitados en el formulario		
5	Se presiona el botón “Guardar”		
DATOS DE ENTRADA			
CAMPO		VALOR	
Nombre	Alfredo		
Apellidos	Perez Sanchez		
Edad	46		
Fecha de Nacimiento	01/09/1974		
Contacto	8390010		
Departamento	Cundinamarca		
Ciudad	Bogotá		
Grupo Étnico	Ninguno		
Nivel Educativo	Educación básica		
Actividad Económica	Alojamiento y Servicios de Comida		
Género	Masculino		
Tipo de Documento	CC		
Número de Identificación	73.851.264		
Correo electrónico	jorivalope@hotmail.com		
Dirección	Calle 58 sur #87-12		
c. Post condiciones			
<ul style="list-style-type: none"> • La información del vendedor informal nuevo, se almacena en la aplicación. • Se envía correo electrónico al vendedor informal, informando sobre el registro y sobre que se autoriza el uso y resguardo de la información ingresada, aplicando la política de protección de datos personales. 			
2. RESULTADOS DE LA PRUEBA			
Defectos y desviaciones			Veredicto
No se presentaron defectos ni desviaciones durante la ejecución de la prueba			PASÓ

Tabla 16 - Caso de Prueba Crear Vendedor

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
CASO DE PRUEBA No.	CP_02	VERSIÓN DE EJECUCIÓN	V1

CASO DE USO:	CU_02	MÓDULO DEL SISTEMA	Gestión de Vendedores
Objetivo del caso de prueba:	Validar que se puede actualizar correctamente la información de un vendedor ya registrado en la aplicación.		
1.CASO DE PRUEBA			
a. Precondiciones			
<ul style="list-style-type: none"> • Contar con un usuario con perfil Administrador ya creado en la aplicación. • Existe información previa sobre vendedor informal registrado en la aplicación. 			
b. Pasos de la prueba			
1	En la vista inicial se visualiza la lista de vendedores informales registrados.		
2	Seleccionar la opción FILTROS		
3	Seleccionar en el campo lista ACTION NEEDED, el valor NOMBRE		
4	Seleccionar en el siguiente campo lista el valor CONTIENE		
5	Ingresar en el tercer campo que se habilita para digitar un valor, el valor ALFREDO		
6	Se presiona el botón APLICAR		
7	Se filtra la lista inicial, dejando solo los registros que cumplen con la condición de búsqueda		
8	Se hace click sobre el registro de interés		
9	Se despliega el formulario con los datos del registro seleccionado		
10	Presionar el botón EDITAR		
11	Se modificar el campo de interés		
12	Se presiona el botón GUARDAR		
DATOS DE ENTRADA			
CAMPO		VALOR	
Contacto		8465213	
Grupo étnico		Raizal	
c. Post condiciones			
<ul style="list-style-type: none"> • Se actualiza correctamente la información de un vendedor informal. 			
2.RESULTADOS DE LA PRUEBA			
Defectos y desviaciones			Veredicto
No se presentaron defectos ni desviaciones durante la ejecución de la prueba			PASÓ

Tabla 17 - Caso de Prueba Actualizar Vendedor

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
CASO DE PRUEBA No.	CP_03	VERSIÓN DE EJECUCIÓN	V1
CASO DE USO:	CU_03	MÓDULO DEL SISTEMA	Gestión de Vendedores
Objetivo del caso de prueba:	Validar que se puede consultar correctamente (si es necesario), la información de un vendedor ya registrado en la aplicación.		
1.CASO DE PRUEBA			
a. Precondiciones			
<ul style="list-style-type: none"> • Contar con un usuario con perfil Administrador ya creado en la aplicación. • Existe información previa sobre vendedor informal registrado en la aplicación. 			
b. Pasos de la prueba			
1	En la vista inicial se visualiza la lista de vendedores informales registrados.		
2	Seleccionar la opción FILTROS		
3	Seleccionar en el campo lista ACTION NEEDED, el valor NOMBRE		
4	Seleccionar en el siguiente campo lista el valor CONTIENE		
5	Ingresar en el tercer campo que se habilita para digitar un valor, el valor ALFREDO		
6	Se presiona el botón APLICAR		
7	Se filtra la lista inicial, dejando solo los registros que cumplen con la condición de búsqueda		
8	Se selecciona el registro que se desea eliminar		
9	Se hace click sobre la opción ACCION - SUPRIMIR		
10	Se muestra mensaje de confirmación de la eliminación del registro seleccionado		
11	Se hace click sobre el botón ACEPTAR		
DATOS DE ENTRADA			
CAMPO		VALOR	
Nombre		Alfredo	
c. Post condiciones			
<ul style="list-style-type: none"> • La información del vendedor ha sido eliminada de la aplicación. 			
2.RESULTADOS DE LA PRUEBA			
Defectos y desviaciones			Veredicto
No se presentaron defectos ni desviaciones durante la ejecución de la prueba			PASÓ

Tabla 18 - Caso de Prueba Eliminar Vendedor

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
CASO DE PRUEBA No.	CP_04	VERSIÓN DE EJECUCIÓN	V1
CASO DE USO:	CU_04	MÓDULO DEL SISTEMA	Gestión de Vendedores
Objetivo del caso de prueba:	Validar que se puede consultar la información de un vendedor ya registrado en la aplicación.		
1.CASO DE PRUEBA			
a. Precondiciones			
<ul style="list-style-type: none"> • Contar con un usuario con perfil de consulta ya creado en la aplicación. • Existe información previa sobre vendedor informal registrado en la aplicación. 			
b. Pasos de la prueba			
1	En la vista inicial se visualiza la lista de vendedores informales registrados.		
2	Seleccionar la opción FILTROS		
3	Seleccionar en el campo lista ACTION NEEDED, el valor NIVEL EDUCATIVO		
4	Seleccionar en el siguiente campo lista el valor ES		
5	Ingresar en el tercer campo que se habilita para digitar un valor, el valor EDUCACION MEDIA		
6	Se presiona el botón AÑADIR UNA CONDICION		
7	Seleccionar en el campo lista ACTION NEEDED, el valor NIVEL EDUCATIVO		
8	Seleccionar en el siguiente campo lista el valor ES		
9	Ingresar en el tercer campo que se habilita para digitar un valor, el valor NEGROS, PALENQUEROS Y RAIZALES		
10	Se presiona el botón APLICAR		
11	Se filtra la lista inicial, dejando solo los registros que cumplen con la condición de búsqueda		
DATOS DE ENTRADA			
DATO		VALOR	
Texto de búsqueda: Nivel Educativo		Educación Media	
c. Post condiciones			
<ul style="list-style-type: none"> • Se muestra mensaje informando que se eliminó correctamente la información de un vendedor informal nuevo. • La información del vendedor ha sido eliminada de la aplicación. 			
2.RESULTADOS DE LA PRUEBA			
Defectos y desviaciones			Veredicto
No se presentaron defectos ni desviaciones durante la ejecución de la prueba			PASÓ

Tabla 19 - Caso de Prueba Consultar Vendedor

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA			
CASO DE PRUEBA No.	CP_05	VERSIÓN DE EJECUCIÓN	V1
CASO DE USO:	CU_05	MÓDULO DEL SISTEMA	Gestión de Vendedores
Objetivo del caso de prueba:	Validar que se puede generar y exportar un reporte con la información de uno o varios vendedores ya registrados en la aplicación.		
1.CASO DE PRUEBA			
a. Precondiciones			
<ul style="list-style-type: none"> • Contar con un usuario con perfil de Administrador ya creado en la aplicación. • Existe información previa sobre vendedor informal registrado en la aplicación. 			
b. Pasos de la prueba			
1	En la vista inicial se visualiza la lista de vendedores informales registrados.		
2	Seleccionar la opción FILTROS		
3	Seleccionar en el campo lista ACTION NEEDED, el valor NIVEL EDUCATIVO		
4	Seleccionar en el siguiente campo lista el valor ES		
5	Ingresar en el tercer campo que se habilita para digitar un valor, el valor EDUCACION MEDIA		
6	Se presiona el botón AÑADIR UNA CONDICION		
7	Seleccionar en el campo lista ACTION NEEDED, el valor GRUPO ETNICO		
8	Seleccionar en el siguiente campo lista el valor ES		
9	Ingresar en el tercer campo que se habilita para digitar un valor, el valor NEGROS, PALENQUEROS Y RAIZALES		
10	Se presiona el botón APLICAR		
11	Se filtra la lista inicial, dejando solo los registros que cumplen con la condición de búsqueda		
12	Se presiona el botón DESCARGAR.		
13	Se genera un archivo en formato Excel con el contenido de la consulta		
DATOS DE ENTRADA			
DATO		VALOR	
Texto de búsqueda: Nivel Educativo		Educación Media	
Texto de búsqueda: Grupo Étnico		Negros, Palenqueros y Raizales	
c. Post condiciones			
<ul style="list-style-type: none"> • Se genera el reporte en formato Excel. 			
2.RESULTADOS DE LA PRUEBA			
Defectos y desviaciones			Veredicto
No se presentaron defectos ni desviaciones durante la ejecución de la prueba			PASÓ

Tabla 20 - Caso de Prueba Generar Reporte

8.3 Escenario de calidad de Mantenibilidad

Para dar cumplimiento al atributo de calidad de Mantenibilidad se planteó el siguiente escenario:

ESCENARIO DE CALIDAD #	8	STAKEHOLDER: Equipo de desarrollo	PRIORIDAD: Alta
Atributo de Calidad	Mantenibilidad		
Justificación	El poder realizar ajustes y mejoras a funcionalidades actuales, o adicionar funcionalidades nuevas, requiere que el código cuente con un nivel de estandarización y baja complejidad de codificación donde la cantidad de “Code Smells” sea el más bajo posible.		
Fuente	Equipo de desarrollo		
Estímulo	El desarrollador ingresa al código fuente para realizar mejoras, correcciones o ampliaciones al código actual.		
Artefacto	Aplicación		
Ambiente	La aplicación se encuentra en operación normal		
Respuesta	El 99.9% de las veces que se realizan ajustes, mejoras o ampliaciones Se utilizan mecanismos y/o herramientas de verificación de incidencias de código, que ayudan a identificar malas prácticas de codificación, y permitan su corrección.		
Medida de la Respuesta	El 1% de incidencias encontradas en el código		

Tabla 21 - Escenario de Calidad para atributo de Mantenibilidad

Para dar cumplimiento al escenario planteado, se realizó la instalación de la extensión SONAR LINT sobre el entorno de desarrollo VISUAL STUDIO CODE. Sonar Lint permite encontrar incidencias, errores, bugs o vulnerabilidades conocidas, de manera automática mientras se escribe código de programación, y presenta explicaciones sobre las mismas como posibles soluciones.

En la imagen que se presenta a continuación, se muestra la integración de Sonar Lint:

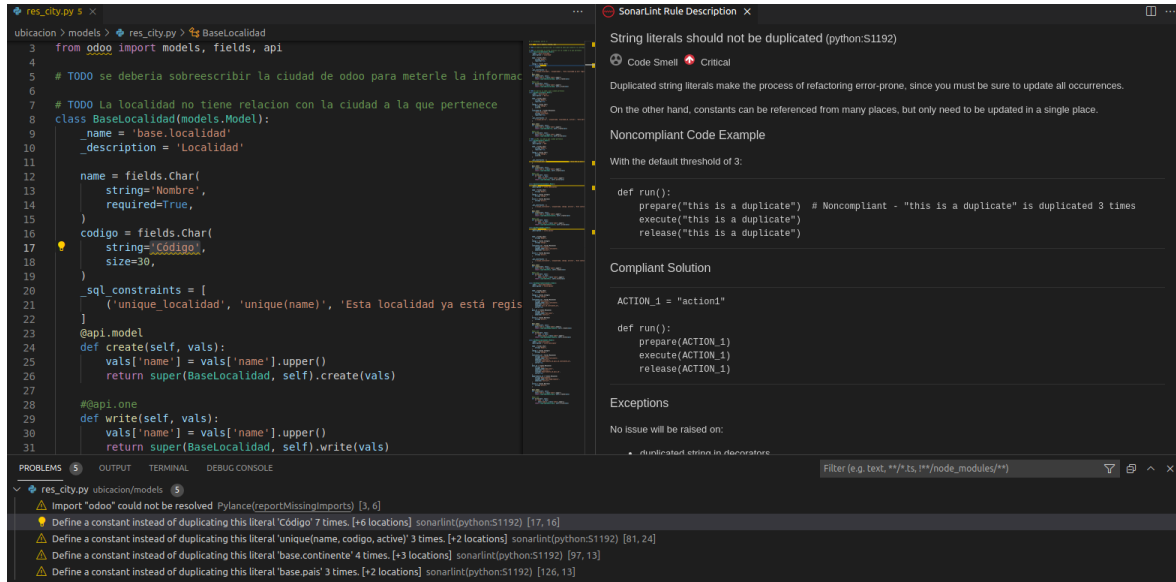


Ilustración 17 - Integración de Sonar Lint a Visual Studio Code (Entorno de Desarrollo) – Fuente propia

Al ejecutar SonarLint, se identifican malos olores (Code Smells), como por ejemplo valores constantes que se duplican en varias líneas de código, o variables que usan, pero no han sido definidas. La imagen a continuación muestra ejemplos de lo descrito:

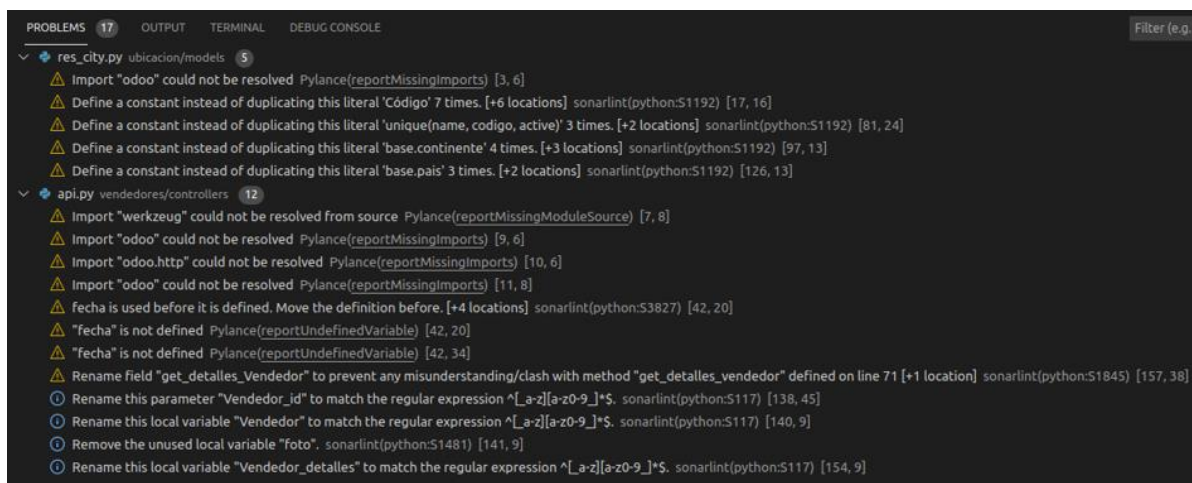


Ilustración 18 - Errores de codificación detectados por Sonar Lint - Fuente propia

9. INSTALACION Y CONFIGURACION

A continuación, se describe los pasos a seguir para realizar la instalación y configuración de los ambientes de producción para el sistema de información para el registro y control de vendedores informales en los municipios de Colombia.

Para el caso del proyecto, se realiza la instalación de todos los componentes que soportan a ODOO y por ende a la aplicación en un solo servidor, el cual cuenta con LINUX UBUNTU 20.04 LTS

Paso 1. Realizar actualización al servidor

- 1.1.sudo apt update
- 1.2.sudo apt upgrade

Paso 2. Se recomienda crear un usuario en Linux para ODOO

- 2.1. sudo adduser -system -home=/opt/odoo -group odoo

Paso 3. Se instala la base de datos que soporta toda la persistencia de ODOO y de las aplicaciones desarrolladas usando ODOO, que es Postgresql

- 3.1. sudo apt-get install postgresql -y

Paso 4. Se crea un usuario que tenga permisos de conexión y administración sobre Postgresql

- 4.1. sudo su - postgres -c "createuser -s odoo" 2> /dev/null || true

Paso 5. Se debe instalar todas las librerías y dependencias de Python.

- 5.1. sudo apt-get install git python3 python3-pip build-essential wget python3-dev python3-venv python3-wheel libxslt-dev libzip-dev libldap2-dev libsasl2-dev python3-setuptools node-less libjpeg-dev gdebi -y

5.2. sudo apt-get install libpq-dev python-dev libxml2-dev libxslt1-dev libldap2-dev libsass2-dev libffi-dev

5.3. sudo -H pip3 install -r
<https://raw.githubusercontent.com/odoo/odoo/master/requirements.txt>

Paso 6. Se instalan otros paquetes necesarios para la correcta ejecución de ODOO

6.1. sudo apt-get install nodejs npm -y
6.2. sudo npm install -g rtlcss
6.3. sudo apt-get install xfonts-75dpi
6.4. sudo wget https://github.com/wkhtmltopdf/packaging/releases/download/0.12.6-1/wkhtmltox_0.12.6-1.bionic_amd64.deb
6.5. sudo dpkg -i wkhtmltox_0.12.6-1.bionic_amd64.deb
6.6. sudo cp /usr/local/bin/wkhtmltoimage /usr/bin/wkhtmltoimage
6.7. sudo cp /usr/local/bin/wkhtmltopdf /usr/bin/wkhtmltopdf

Paso 7. Se crean el directorio donde se ubican los archivos de log.

7.1. sudo mkdir /var/log/odoo
7.2. sudo chown odoo:odoo /var/log/odoo

Paso 8. Se accede al repositorio de GIT desde donde se descargan los instaladores de ODOO

8.1. sudo apt-get install git

Paso 9. Se precede a clonar e instalar ODOO

9.1. sudo git clone --depth 1 --branch master <https://www.github.com/odoo/odoo>
/odoo/odoo-server

Paso 10. Se asignan permisos al directorio de ODOO

10.1. sudo chown -R odoo:odoo /odoo/*

Paso 11. Se crea el archivo de configuración

11.1. sudo touch /etc/odoo-server.conf
11.2. sudo su root -c "printf '[options] \n; This is the password that allows database operations:\n' >> /etc/odoo-server.conf"
11.3. sudo su root -c "printf 'admin_passwd = admin\n' >> /etc/odoo-server.conf"
11.4. sudo su root -c "printf 'xmlrpc_port = 8069\n' >> /etc/odoo-server.conf"

11.5. sudo su root -c "printf 'logfile = /var/log/odoo/odoo-server.log\n' >> /etc/odoo-server.conf"

11.6. sudo su root -c "printf 'addons_path=/odoo/odoo-server/addons\n' >> /etc/odoo-server.conf"

11.7. sudo chown odoo:odoo /etc/odoo-server.conf

11.8. sudo chmod 640 /etc/odoo-server.conf

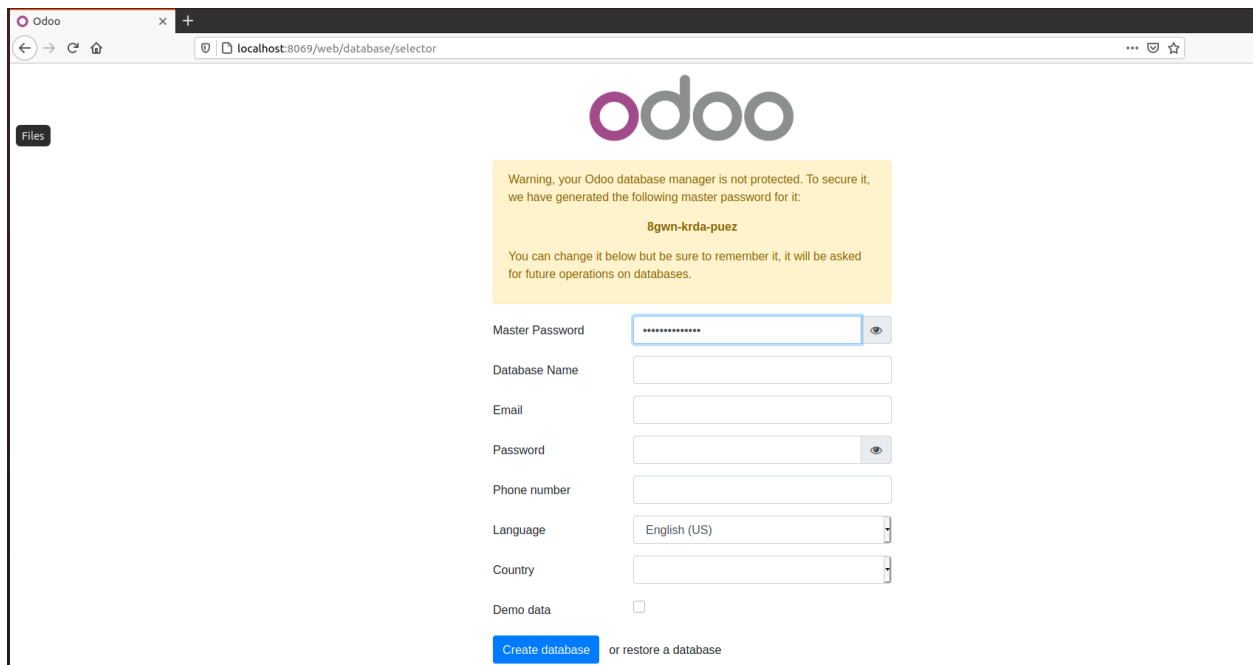
Paso 12. Se inicia ODOO

12.1. sudo su - odoo -s /bin/bash

12.2. cd /odoo/odoo-server

12.3. ./odoo-bin -c /etc/odoo-server.conf

Paso 13. Se accede a la interfaz web de ODOO para comprobar que ya está operativo.



The screenshot shows a web browser window with the URL `localhost:8069/web/database/selector`. The page features the Odoo logo at the top center. Below the logo is a yellow warning box that reads: "Warning, your Odoo database manager is not protected. To secure it, we have generated the following master password for it: **8gwn-krda-puez**". Below the warning, there is a form with the following fields: "Master Password" (with a masked input and an eye icon), "Database Name", "Email", "Password" (with a masked input and an eye icon), "Phone number", "Language" (set to "English (US)"), "Country", and "Demo data" (with an unchecked checkbox). At the bottom of the form, there is a blue button labeled "Create database" followed by the text "or restore a database".

Ilustración 19 - Formulario para la creación de usuario administrador y base de datos de Odoo – Fuente propia

En esta pantalla se realiza la creación del usuario administrador y de la base de datos que va a soportar el sistema de información.

Paso 15. Instalación de nginx

15.1 sudo apt install nginx

Paso 16. Se habilita a nginx en el firewall

16.1 sudo ufw allow 'Nginx HTTP'

Paso 17. Se crea el bloque en el servidor para habilitar un dominio

17.1 sudo mkdir -p /var/www/your_domain/html

17.2 sudo chown -R \$USER:\$USER /var/www/your_domain/html

17.3 sudo chmod -R 755 /var/www/your_domain

17.4 sudo nano /etc/nginx/sites-available/default

Paso 18. Se procede a instalar el leaflet

18.1 sudo apt-get install -y libjs-leaflet

Paso 19. Posterior a la creación del usuario administrador y de la base de datos, el ODOO dispone de una pantalla para el ingreso al esquema creado.

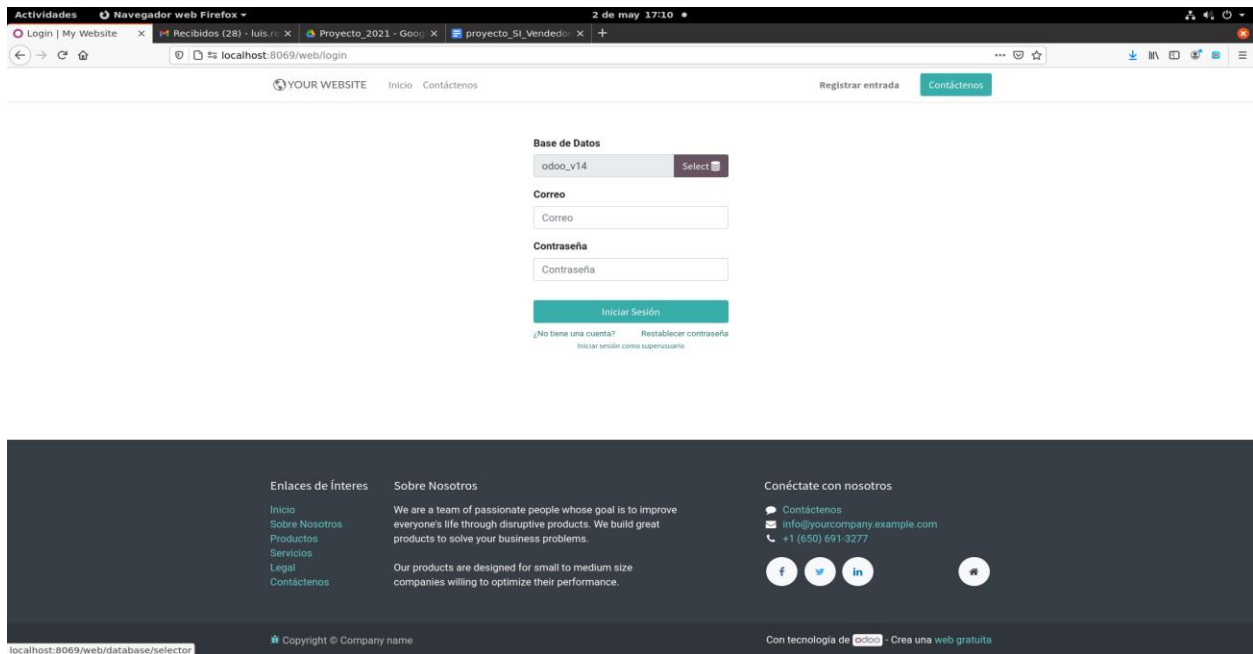


Ilustración 20 - Pantalla para ingreso a Odoo - Fuente propia

Paso 20. Al ingresar con el usuario administrador, se visualizan las aplicaciones que han sido desarrolladas y al que el usuario registrado tiene acceso, para su administración.

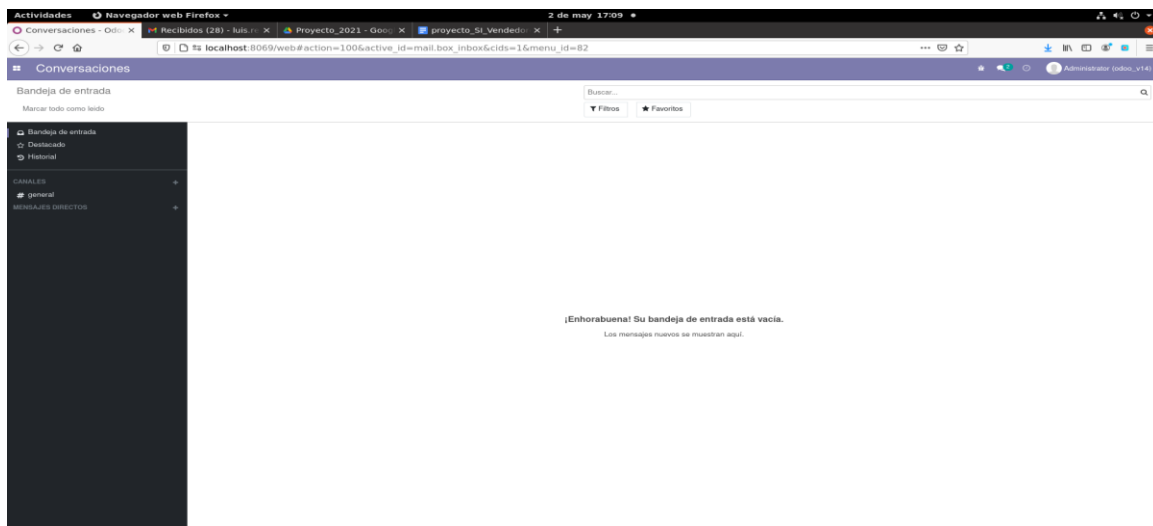


Ilustración 21 - Pantalla de visualización de aplicaciones asociadas al usuario administrador - Fuente propia

Paso 21. Para agregar una aplicación nueva, se debe ir a la opción CONVERSACIONES y allí seleccionar APLICACIONES.

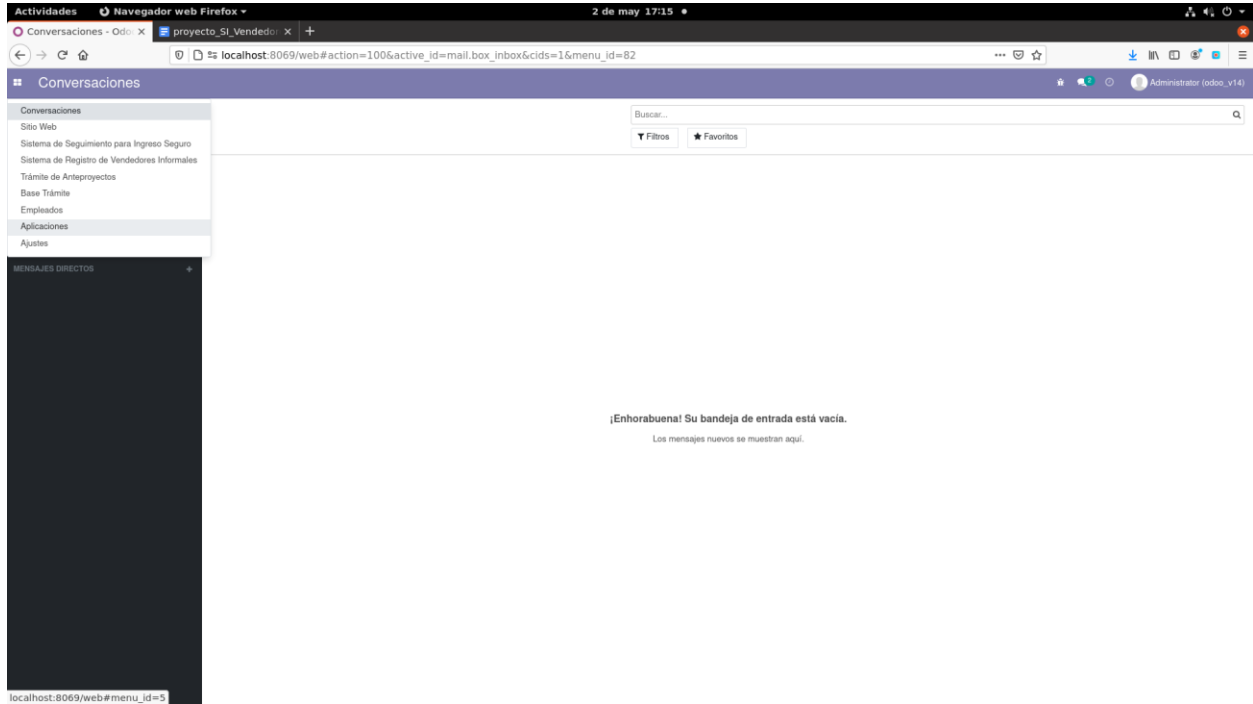


Ilustración 22 - Opción para búsqueda de aplicaciones ya existentes en Odoo o creadas

Paso 22. Se visualizan todas las posibles aplicaciones.

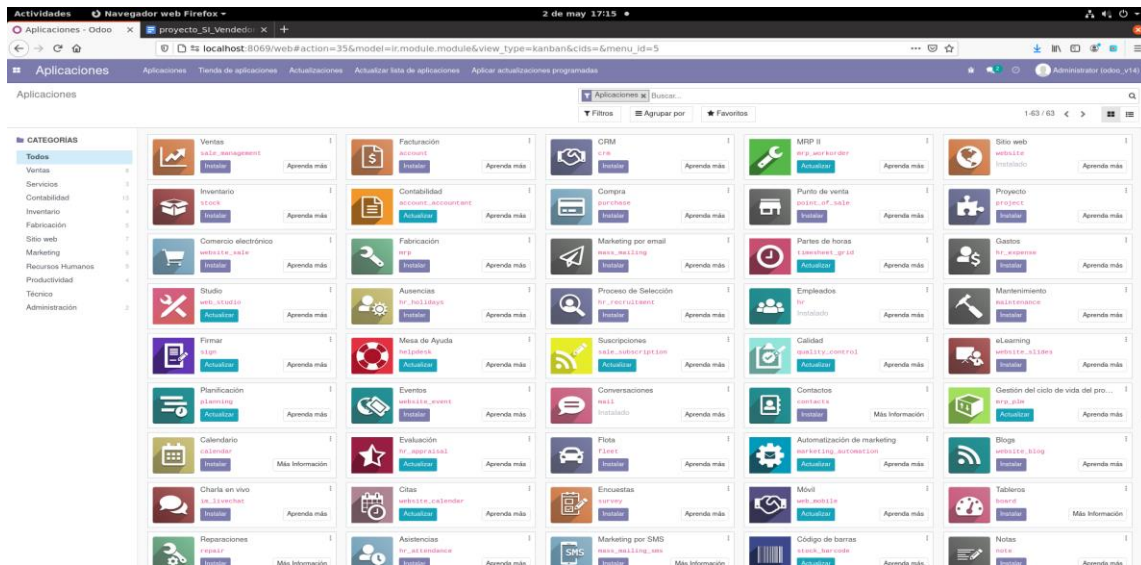


Ilustración 23 - Lista de aplicaciones existentes y registradas en Odoo

Paso 23. En caso que el módulo o aplicación nuevo no este registrada en Odoo, se debe adicionar a través de la opción Actualizar Módulo.

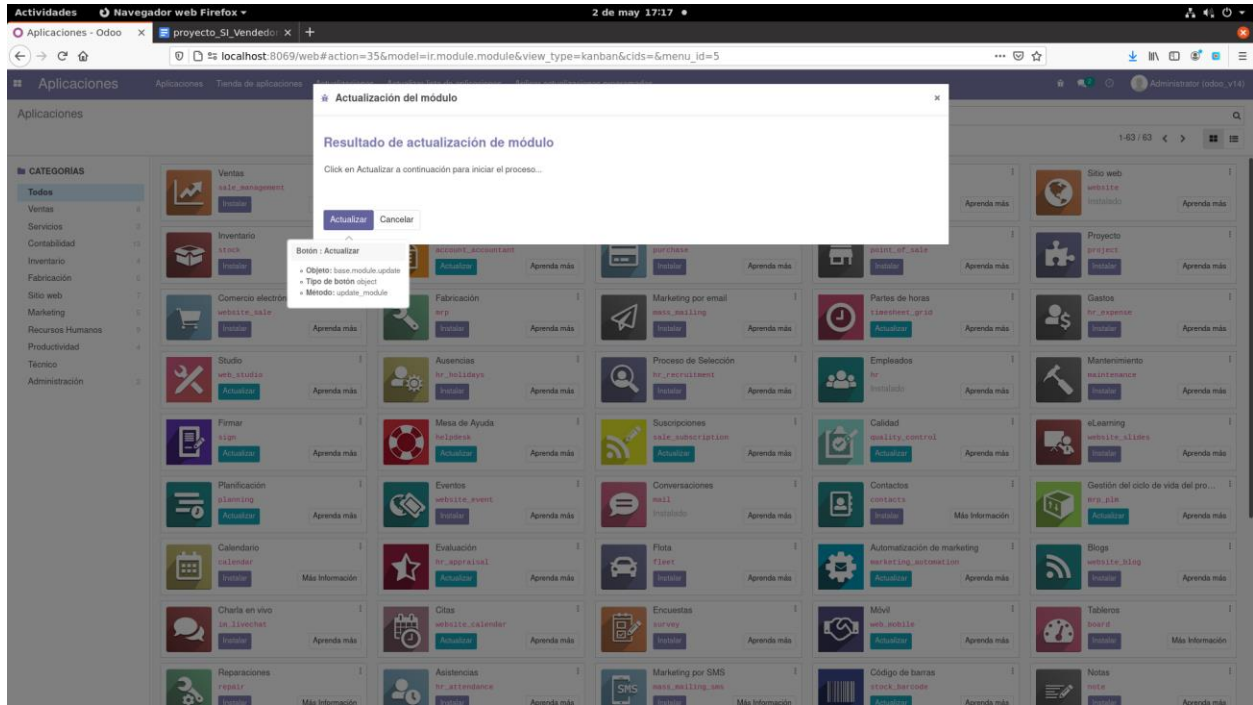


Ilustración 24 - Actualización de módulos en Odoo - Fuente propia

Paso 24. Se visualizan los módulos nuevos que fueron cargados al servidor de aplicaciones de Odoo.

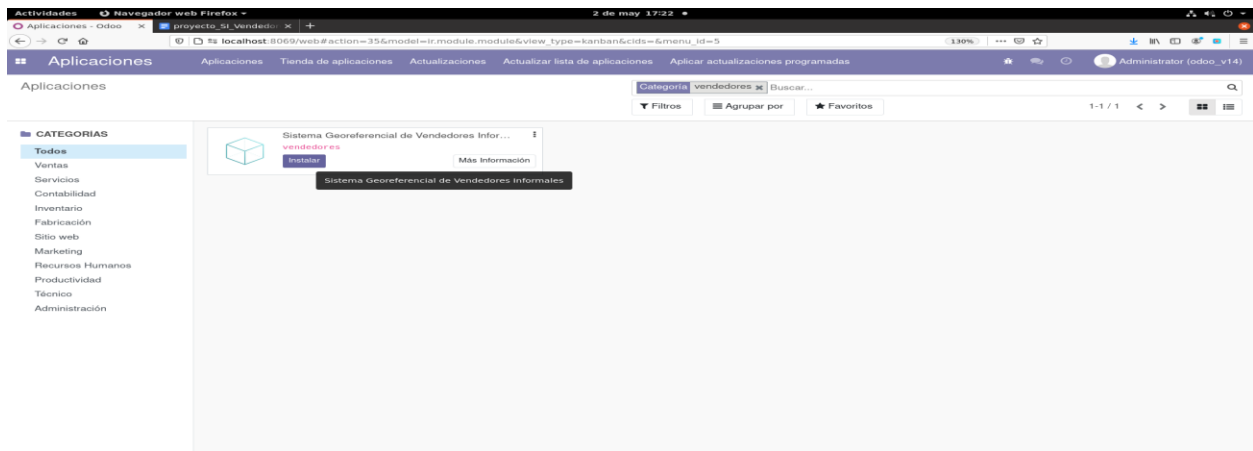


Ilustración 25 - Carga de nuevas aplicaciones a Odoo - Fuente Propia

Paso 25. Se debe asignar permisos a los usuarios para que puedan acceder a la aplicación cargada a Odoo.

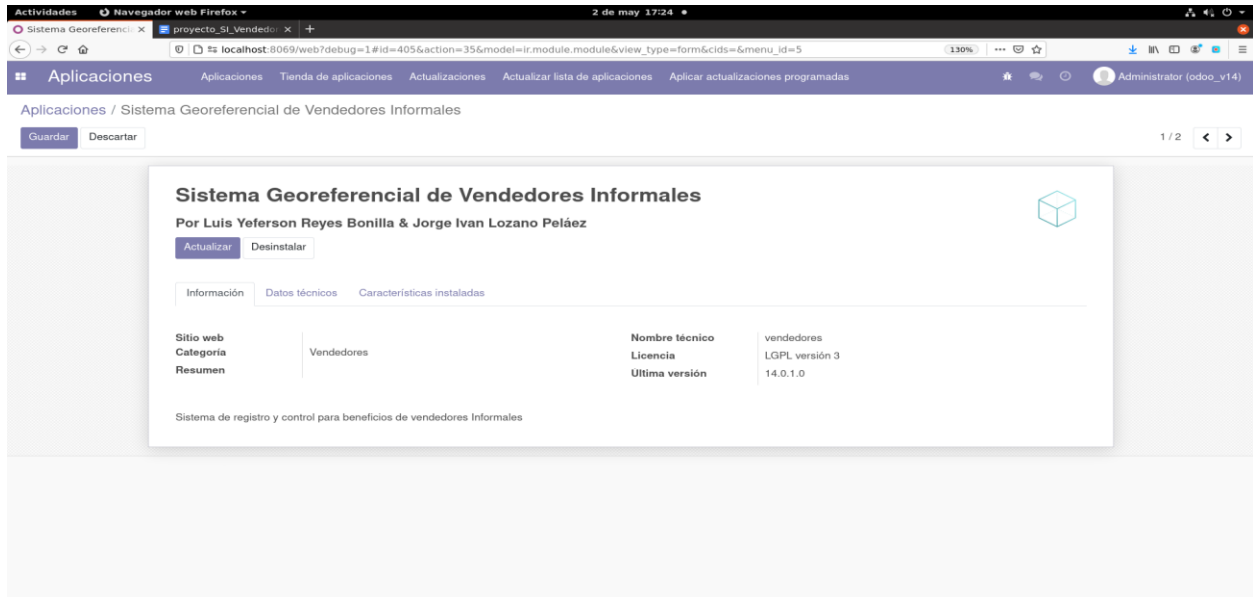


Ilustración 26 - Identificación de aplicación a asignar permisos - Fuente propia

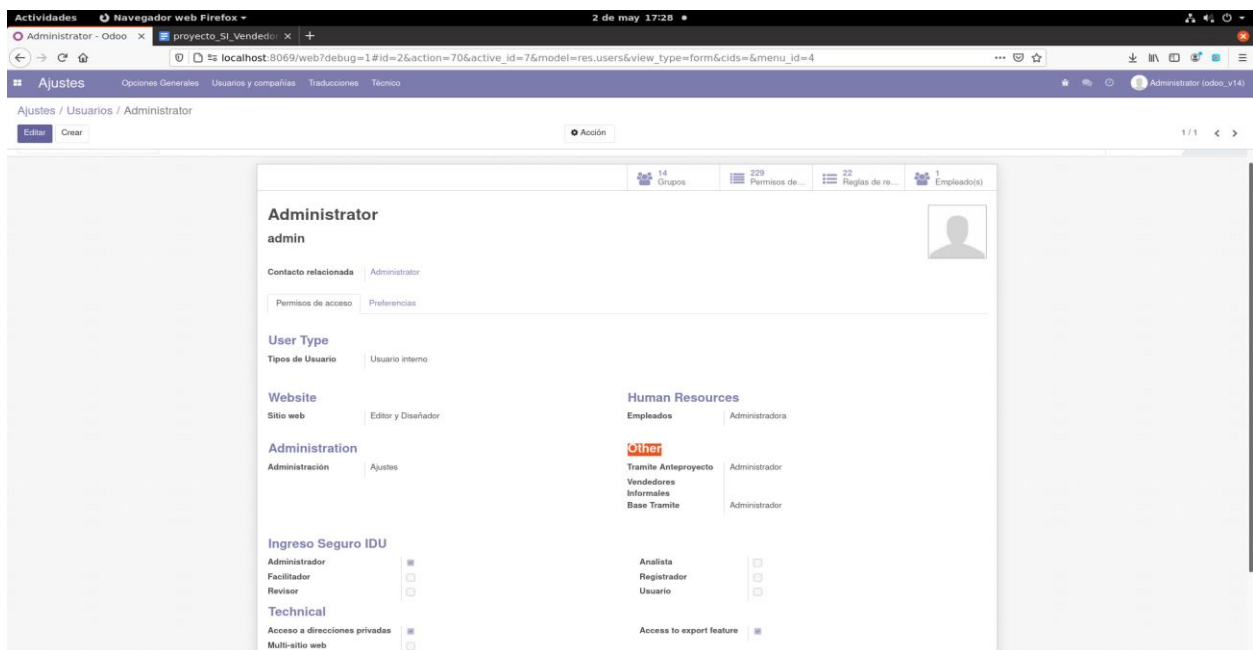


Ilustración 27 - Se identifica a que aplicación se debe asignar de permisos a usuarios poder ejecutar la aplicación

- Fuente propia

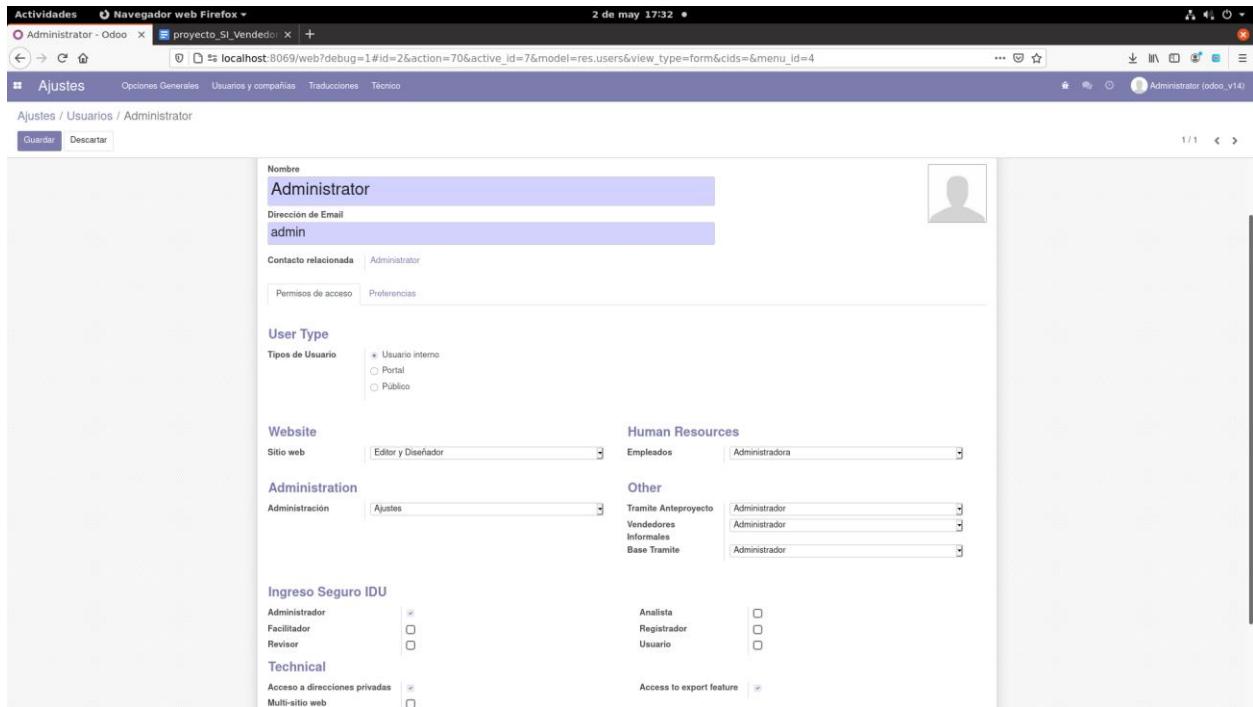


Ilustración 28 - Asignación de permisos sobre la aplicación de interés - Fuente propia

Paso 26. Se realiza nuevamente la búsqueda de la aplicación recién cargada a Odoo.

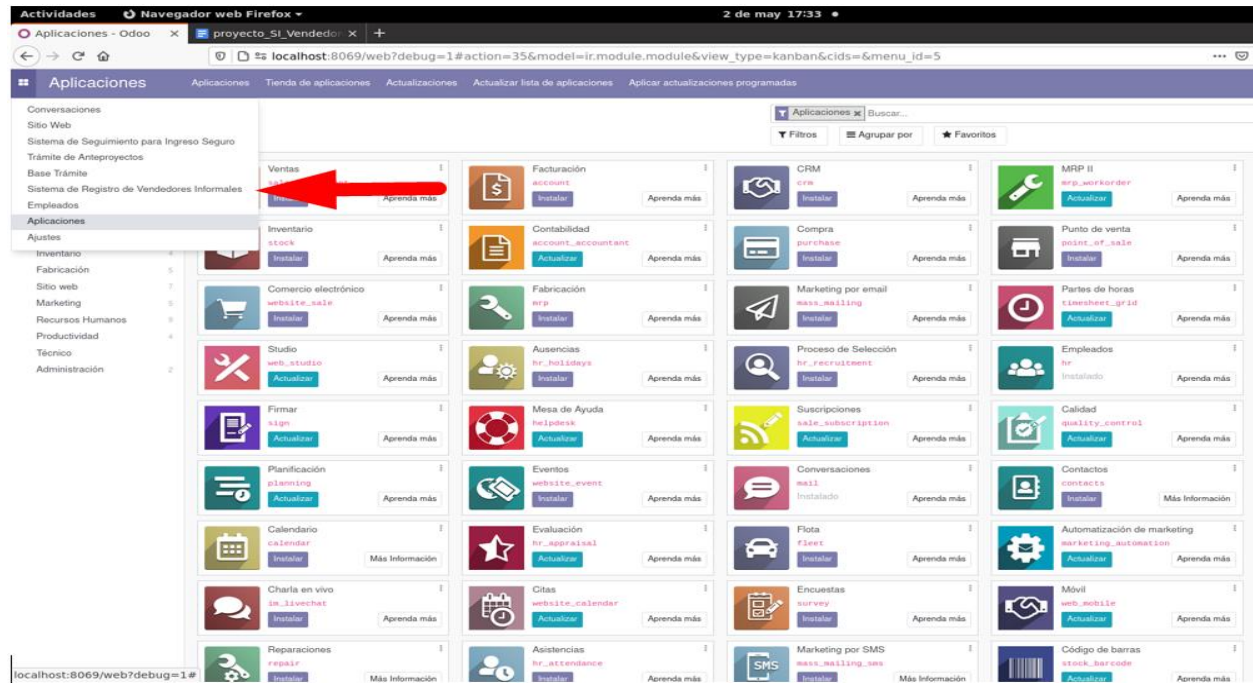


Ilustración 29 - Se valida que la nueva aplicación esta ya cargada en el entorno de Odoo - Fuente propia

Paso 27. Se valida que se tenga acceso a la nueva aplicación cargada.

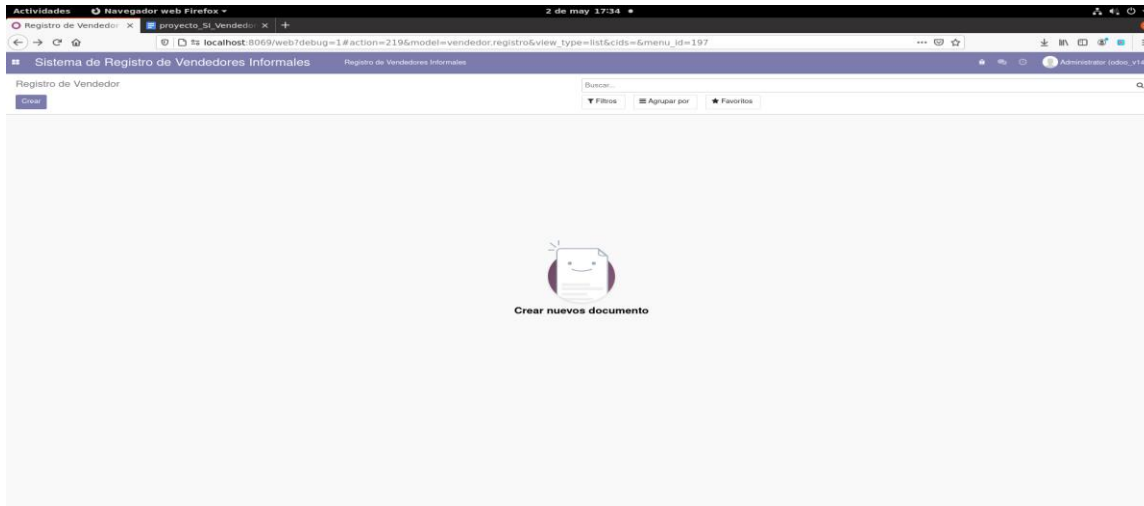


Ilustración 30 - Validación de acceso a aplicación - Fuente propia

CONCLUSIONES

- Se logró desarrollar una aplicación web que facilita y permite el registro y consulta de información básica de los vendedores informales, para cualquier municipio que la use y que se responde a los lineamientos generales de la Política Pública de Vendedores Informales.
- El punto anterior valida que el desarrollo Web sigue siendo uno de los mejores enfoques, para la creación de sistemas de información de todo tamaño y complejidad, pero sobre todo que se prestan para poder llegar a un gran número de interesados.
- Se definió una arquitectura MVC para la construcción de la solución planteada, la que facilitó que el resultado final ofrezca una estructura que permita su mantenibilidad en el tiempo, aprovechando las ventajas de segmentación funcional, donde cada capa se puede manejar como un desarrollo independiente que facilita su mantenimiento y pruebas a cada capa.
- Se tuvo presente aplicar las recomendaciones generales de la política de protección y manejo de datos dictadas por el gobierno nacional (ley 1581 de 2012).
- Se construyeron y validaron escenarios de calidad, asociados a los atributos de calidad de mantenibilidad y de adecuación funcional, aplicando herramientas como SONAR LINT para el primer atributo, y casos de pruebas para el segundo atributo.

- Se evidenció que el uso de metodologías ágiles como lo es KANBAN, beneficia los procesos de desarrollo de software cuando el tiempo de desarrollo es un factor determinante, asociado a la calidad, al permitir lograr una documentación apropiada, entregables funcionales e incrementales y el cumplimiento de las expectativas de los interesados.

REFERENCIAS

Aguilar Avilés, C. R. (2018). APP de ventas móviles que permite conectar a clientes con vendedores ambulantes (Tesis de pregrado). Universidad de las Américas, Quito.

<http://dspace.udla.edu.ec/handle/33000/10006>

Diseño web adaptable.

[https://es.wikipedia.org/wiki/Dise%C3%B1o_web_adaptable#:~:text=El%20dise%C3%B1o%20web%20adaptable%20\(tambi%C3%A9n,se%20est%C3%A1%20utilizando%20para%20visitarlas.](https://es.wikipedia.org/wiki/Dise%C3%B1o_web_adaptable#:~:text=El%20dise%C3%B1o%20web%20adaptable%20(tambi%C3%A9n,se%20est%C3%A1%20utilizando%20para%20visitarlas.)

Instituto para la economía social IPES. <http://www.ipes.gov.co/>

Mercadolocal. <https://www.mercadolocal.co/>

Ministerio del interior, Ministerio de Trabajo. (2020), Política pública de vendedores informales.

PostgreSQL. <https://es.wikipedia.org/wiki/PostgreSQL>

Python. <https://es.wikipedia.org/wiki/Python>

QGIS. <https://es.wikipedia.org/wiki/QGIS>

Registro Individual de Vendedores (RIVI).

<https://soachainiciativaciudadana.com/conozca-el-formulario-online-para-el-registro-de-vendedores-informales/>

República de Colombia (2019). Ley 1988 del 2 de agosto de 2019

Schwaber Ken y Sutherland Jeff (2016). La Guía Definitiva de Scrum: Las Reglas del Juego

Una app para poner en contacto a vendedores ambulantes con los pueblos, ganadora del hackaton por la España Vacía de HP.

<https://www.europapress.es/portaltic/software/noticia-app-poner-contacto-vendedores-ambulantes-pueblos-ganadora-hackaton-espana-vacia-hp-20191113154634.html>

apd. ¿En qué consiste la metodología Kanban y cómo utilizarla?

<https://www.apd.es/metodologia-kanban/#:~:text=La%20metodolog%C3%ADa%20Kanban%20es%20un,c%C3%B3mo%20se%20completan%20las%20areas.>

Metodología Kanban. <https://iswugkanbansite.wordpress.com/inicio/roles/>

sahu, shivam. Technical Architecture of Odoo and Deployment Structure at Server.

<https://erpsolutions.oodles.io/blog/odoo-architecture-technical-deployment/>

technaureus. MVC Architecture of Odoo.

<https://www.technaureus.com/mvc-architecture-of-odoo/>

Blancarte, O. (2020), Introducción a la Arquitectura de Software, un enfoque práctico (Primera edición). Ciudad de México, México: Centroamerica.

Bass, L., Clements, P., & Kazman, R. (2015). Software Architecture in Practice (Third Edition). Massachussets. United States: Addison-Wesley.

Sonarlint. <https://www.sonarlint.org/features/>