

ESCUELA DE INGENIERÍA EN SISTEMAS

Tema:

DESARROLLO DE UNA PLATAFORMA *ONLINE* PARA PROMOCIÓN DE SERVICIOS PERSONALES Y PROFESIONALES

Proyecto de Investigación previo a la obtención del título de Ingeniero en Sistemas y Computación

Línea de Investigación:

SISTEMAS DE INFORMACIÓN Y/O NUEVAS TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN Y SUS APLICACIONES

Autor:

Alex Napoleón Espinoza Sánchez

Director:

Mg. José Marcelo Balseca Manzano

Ambato - Ecuador

Septiembre 2019

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE AMBATO

HOJA DE APROBACIÓN

Tema:

Desarrollo de una plataforma *online* para promoción de servicios personales y profesionales.

Línea de Investigación:

Sistemas de Información y/o nuevas tecnologías de la información y comunicación y sus aplicaciones

Autor:

Alex Napoleón Espinoza Sánchez

José Marcelo Balseca Manzano, Ing. Mg. f. _____

CALIFICADOR

Acurio Maldonado Santiago Alejandro, Ing. Mg. f. _____

CALIFICADOR

Garcés Freire Enrique Xavier, Ing. Mg. f. _____

CALIFICADOR

Mónica Patricia Mena Moreno, Ing. Mg. f. _____

DIRECTORA DE LA ESCUELA DE INGENIERÍA EN SISTEMAS

Hugo Rogelio Altamirano Villarroel, Dr. f. _____

SECRETARIO GENERAL PUCESA

Pontificia Universidad Católica del Ecuador
SECRETARIA GENERAL
PROCURADURIA

Ambato – Ecuador

Septiembre 2019

Pontificia Universidad Católica del Ecuador
BIBLIOTECA

DECLARACIÓN DE AUTENCIDAD Y RESPONSABILIDAD

Yo: ALEX NAPOLEÓN ESPINOZA SÁNCHEZ, con CC. 171722703-5, autora del trabajo de graduación intitulado: "DESARROLLO DE UNA PLATAFORMA *ONLINE* PARA PROMOCIÓN DE SERVICIOS PERSONALES Y PROFESIONALES", previa a la obtención del título profesional de INGENIERO DE SISTEMAS Y COMPUTACIÓN, en la escuela de INGENIERÍA DE SISTEMAS

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE Ambato, el referido trabajo de graduación, respetando las políticas de propiedad intelectual de Universidad

Ambato, septiembre 2019

Pontificia Universidad Católica del Ecuador
BIBLIOTECA

ALEX NAPOLEÓN ESPINOZA SÁNCHEZ

CC. 171722703-5

AGRADECIMIENTO

En primera instancia agradezco a DIOS, porque en el encontré la fortaleza para no desistir de este gran anhelo, agradezco también a mis catedráticos, que inculcaron en mi toda su sabiduría y me han ayudado paso a paso a llegar al punto en el que hoy me encuentro.

El camino no ha sido sencillo, pero gracias a las ganas de transmitirme sus conocimientos, he logrado llegar a obtener importantes metas como lo es el culminar el desarrollo de mi tesis y con mucha alegría obtener mi título profesional como Ingeniero.

Atentamente

Alex N. Espinoza S.

DEDICATORIA

Dedico esta Tesis a mis Padres Rita Sánchez y Napoleón Espinoza, quienes siempre me apoyaron incondicionalmente de manera moral como económica, para poder llegar a obtener mi título profesional.

A mi hermana, sobrino, cuñado y demás familia en general que me brindaron sus palabras de aliento en el transcurso de mi carrera Universitaria.

Alex N. Espinoza S.

RESUMEN

El desarrollo de una plataforma *online* para promoción de servicios personales y profesionales tiene como objetivo reducir o eliminar la falta de información para contactar y localizar un servicio para realizar un trabajo específico, con la creación de una plataforma *online* que facilite la búsqueda de servicios personales o profesionales, permitiendo al público en general acceder a una base de datos en donde podrán geolocalizar y obtener un servicio acorde a sus necesidades, el cual podrán usar desde cualquier dispositivo con acceso a internet además de contar con diseño responsivo para mayor comodidad en su uso. Se solventa la necesidad de búsqueda rápida y practica de un servicio requerido

El método aplicado fue el *Object Oriented Hipermedia Design Method* (OOHDM) que se usó para el desarrollo de proyecto *web*, se detallan las acciones que se llevan a cabo en las 5 etapas que componen el método, se recalcan los semblantes interdisciplinarios de cada una de las fases.

Para finalizar se detalla los procesos y requisitos a cumplir en las etapas para culminar el desarrollo del proyecto dejando en evidencia las conclusiones más relevantes obtenidas en la investigación entre las cuales se pueden resumir puntos como, el método OOHDM permitió la documentación del proceso total de desarrollo de la plataforma, permitiendo usar de mejor forma las cualidades del *framework* seleccionado en conjunto con las aplicaciones proporcionadas con la plataforma de desarrolladores de Google usadas para la geolocalización de los usuarios.

Palabras clave:

Responsive, Geolocalización, Ofertante, Plataforma, *online*.

ABSTRACT

The aim of developing an online platform to advertise personal and professional services is to reduce or eliminate information gaps when a potential customer is looking for a specific service and to grant access to a database with geo-location to find a service that is required from any device connected to the Internet. The platform's design is responsive and easy-to-use to find a service quickly and practically. Object-Oriented Hypermedia Design Methodology (OOHDM) was applied to specify the actions carried out in five stages underlining the interdisciplinary aspects of each stage. In addition, the processes and requirements are detailed in each of the stages, proving that the OOHDM method provided all of the information of the complete development of the platform. This made it possible to optimize the framework's main features with the applications that were provided with Google's developer platform for the user´s geo-location.

Keywords: responsive, geo-location, offeror, platform, online.

ÍNDICE

DECLARACIÓN DE AUTENCIDAD Y RESPONSABILIDAD iii
AGRADECIMIENTO iv
DEDICATORIA v
RESUMEN vi
ABSTRACT vii
ÍNDICE viii
ÍNDICE DE ILUSTRACIONES xii
ÍNDICE DE TABLAS xv
INTRODUCCIÓN 1
Preguntas Básicas 2
Objetivos 3
Objetivo General 3
Objetivos Específicos 3
Delimitación del Proyecto 3
Funcional 3
Temporal 4
Justificación 4
Meta 5
CAPITULO I. ESTADO DEL ARTE Y LA PRÁCTICA 6
1.1. FUNDAMENTOS TEÓRICOS 6
1.1.1. Internet 6
1.1.2. Sitio *web* 6
1.1.3. Diferencia entre sitio *web* y pagina *web* 7
1.2. DESCRIPCÓN DE INGENIERÍA *WEB*, ARQUITECTURA *WEB* Y APLICACIONES *WEB* 7
1.2.1. Ingeniería *Web* (I*Web*) 7
1.2.2. Arquitectura *Web* 8
1.2.3. Aplicaciones *web* 9
1.3. VENTAJAS Y DESVENTAJAS DE LAS APLICACIONES *WEB* 10
1.3.1. Ventajas de las aplicaciones *Web* 10
1.3.2. Desventajas de las aplicaciones *Web* 11

1.4. TIPOS DE APPS *WEB*, TECNOLOGÍAS, ENTORNOS DE EJECUCIÓN E IMPLEMENTACIÓN 11
1.4.1. Tipos de Apps *web* que se pueden desarrollar 11
1.4.2. Tecnologías, entornos de ejecución e implementación 12
1.4.3. Sistema de Notificaciones 13
1.5. HERRAMIENTAS DE EJECUCION Y DESARROLLO 14
1.5.1. Plataforma de Ejecución 14
1.5.2. Herramientas de Redacción y Desarrollo 14
CAPITULO II. DISEÑO METODOLOGICO PARA EL DESARROLLO DE UNA PLATAFORMA ON LINE 15
2.1. MÉTODOS 15
2.1.1. Método OOHDM y sus etapas para la aplicación 15
2.1.2. Primera Etapa. Obtención de Requerimiento 16
2.1.3. Segunda Etapa. Modelo Conceptual 16
2.1.4. Tercera Etapa. Estructura Navegacional 19
2.1.5. Cuarta Etapa. Interfaz Abstracta 22
2.1.6. Quinta Etapa. Implementación 24
2.2. USO DE OOHDM Y SUS DIFERENTES MODELOS 26
2.2.1. Modelos arquitectónicos 27
2.2.2. Modelos navegacionales 27
2.2.3. Modelos de interfaz 27
CAPITULO III. DESARROLLO DEL MODELO E IMPLEMENTACION DE LA PLATAFORMA ON LINE 28
3.1. PRIMERA ETAPA. OBTENCIÓN DE REQUERIMIENTO 28
3.1.1. Tópicos a ser atendidos 28
3.1.2. Tópicos relacionados entre sí 29
3.1.3. Usuarios a ser atendidos 29
3.1.4. Tareas principales a ser abordadas 29
3.1.5. Tareas y categorías de usuarios 30
3.1.6. Diagramas de casos de uso 30
3.2. SEGUNDA ETAPA. MODELO CONCEPTUAL 54
3.2.1. Tbl_carreras 55
3.2.2. Tbl_especialidad 55
3.2.3. Diagrama UML Carrera -Especialidad 55
3.2.4. Tbl_provincias 56

3.2.5. Tbl_ciudades 56
3.2.6. Diagrama UML Provincia-Ciudad 56
3.2.7. Tbl_usuarios 57
3.2.8. Tbl_ofertantes 57
3.2.9. Diagrama UML Usuarios - Ofertantes 58
3.2.10. Tbl_noofertantes 59
3.2.11. Diagrama UML Usuarios - Noofertantes 59
3.2.12. Tbl_contratos 60
3.2.13. Diagrama UML Contratos - ofertantes y noofertante 60
3.3. TERCERA ETAPA. ESTRUCTURA NAVEGACIONAL 62
3.3.1. Niveles usuarios ofertantes 62
3.3.2. Niveles usuarios no ofertantes 66
3.3.3. Nivel de Administración. 70
3.3.4. Diagrama General Navegacional. 72
3.4. CUARTA ETAPA. INTERFAZ ABSTRACTA 73
3.4.1. Diseño adaptativo o *Responsive Design* 73
3.4.2. *Bootstrap* vs. *Foundation* vs. *Materialize* 74
3.4.3. *Framework Bootstrap* 74
3.4.4. Modulo: Inicio 76
3.4.5. Modulo: Tipos de Registro 78
3.4.6. Modulo. Registro Ofertante / No ofertante 79
3.4.7. Modulo: Inicio de sección 80
3.4.8. Modulo: Editar Perfil 81
3.4.9. Modulo: Perfil de Ofertantes 83
3.4.10. Modulo: Perfil de No ofertantes 84
3.4.11. Modulo: Bienvenido 85
3.4.12. Sección: Cerca de ti 88
3.4.13. Sección: Búsqueda general 89
3.4.14. Módulo: Búsqueda general 89
3.4.15. Sección: Pendientes 91
3.4.16. Sección: Pendientes / Botón de Contacto 92
3.4.17. Sección: Destacados 94
3.4.18. Modulo: Administración de usuarios 95
3.4.19. Modelo entidad / relación 96

3.5. QUINTA ETAPA. IMPLEMENTACIÓN 97
3.5.1. Hosting y Dominio 97
3.5.2. Modelo lógico de la base de datos 98
3.5.3. Geolocalización 103
3.5.4. Sistema de calificación 108
3.5.5. Sistemas de Notificación 108
3.5.6. Funciones de la plataforma 109
3.5.7. Login / inicio de sesión 112
3.5.8. Registrar Usuario 112
3.5.9. Búsqueda 113
3.5.10. Modificar Datos de Usuario 115
3.5.11. Eliminar Usuario 116
3.5.12. Funciones dinámicas 117
3.5.13. Análisis y Validación de los Resultados 119
3.5.14. Certificado *Secure Socket Layer* 123
3.5.15. Problemas encontrados en la implementación 137
CONCLUSIONES 138
RECOMENDACIONES 138
BIBLIOGRAFÍA 139

ÍNDICE DE ILUSTRACIONES

Ilustración 1, Usuario atendidos _____ 31
Ilustración 2, UCD - Usuarios administrador _____ 32
Ilustración 3, UCD - Usuarios _____ 37
Ilustración 4, UCD - Usuarios ofertantes _____ 38
Ilustración 5, UCD – Usuarios no ofertantes _____ 39
Ilustración 6, CRD Modelo conceptual _____ 54
Ilustración 7, CRD Carrera-Especialidad _____ 55
Ilustración 8, CRD Provincia-Ciudad _____ 56
Ilustración 9, CRD Usuarios-Ofertantes _____ 58
Ilustración 10, CRD Usuarios-Noofertantes _____ 59
Ilustración 11, CRD Contratos-Usuarios no Administradores _____ 61
Ilustración 12, NCS Registro Ofertante _____ 62
Ilustración 13, NCS Registro cancelado _____ 63
Ilustración 14, NCS Editar perfil _____ 63
Ilustración 15, NCS Cambiar Contraseña _____ 64
Ilustración 16, NCS Búsqueda y Solicitud _____ 64
Ilustración 17, NCS Atender Solicitud _____ 65
Ilustración 18, NCS Calificar Usuario no Administrador _____ 65
Ilustración 19, NCS Eliminar Solicitud _____ 66
Ilustración 20, NCS Registro no Ofertante _____ 66
Ilustración 21, NCS Registro Cancelado _____ 67
Ilustración 22, NCS Editar Perfil No Ofertante _____ 67
Ilustración 23, NCS Cambiar Contraseña _____ 68
Ilustración 24, NCS Búsqueda y Solicitud _____ 68
Ilustración 25, NCS Calificar Ofertante _____ 69
Ilustración 26, Cancelar Solicitud de Contacto _____ 69
Ilustración 27, NCS Crear Usuario _____ 70
Ilustración 28, NCS Modificar Usuario _____ 70
Ilustración 29, NCS Eliminar Usuario _____ 71
Ilustración 30, Diagrama General Navegacional _____ 72
Ilustración 31, ADV Modulo Inicio _____ 77

Ilustración 32, ADV Modulo Tipos de Registro _____ 78
Ilustración 33, ADV Modulo Registro _____ 79
Ilustración 34, ADV Modulo Inicio de Sesión _____ 80
Ilustración 35, ADV Editar Perfil Ofertante _____ 81
Ilustración 36, ADV Editar Perfil No Ofertante _____ 82
Ilustración 37, ADV Perfil Ofertante _____ 83
Ilustración 38, ADV Perfil No Ofertante _____ 84
Ilustración 39, ADV Modulo Bienvenido - Parte 1 _____ 86
Ilustración 40, ADV Modulo Bienvenido - Parte 2 _____ 87
Ilustración 41, ADV Sección - Cerca de ti _____ 88
Ilustración 42, ADV Sección - Búsqueda General _____ 89
Ilustración 43, Modulo Búsqueda General _____ 90
Ilustración 44, ADV Sección Pendientes - Parte 1 _____ 91
Ilustración 45, ADV Sección Pendientes - Parte 2 _____ 92
Ilustración 46, ADV Sección Pendientes - Contacto _____ 93
Ilustración 47, ADV Sección Destacados _____ 94
Ilustración 48, ADV Administración de Usuarios _____ 95
Ilustración 49, ERD Diagrama Entidad Relación _____ 96
Ilustración 50, Modelo lógico BD _____ 99
Ilustración 51, Tabla Usuarios BD _____ 99
Ilustración 52, Tabla Ofertantes BD, _____ 100
Ilustración 53, Tabla No Ofertantes BD _____ 100
Ilustración 54, Tabla Provincias BD, _____ 101
Ilustración 55, Tabla Ciudades BD, _____ 101
Ilustración 56, Tabla Carreras BD _____ 101
Ilustración 57, Tabla Especialidades BD, _____ 102
Ilustración 58, Tabla Contratos BD _____ 102
Ilustración 59, API KEY Google _____ 103
Ilustración 60, Grafica pastel - Accesos por país - mes más exitoso _____ 119
Ilustración 61, Acceso de los 4 primeros países - mes más exitoso _____ 119
Ilustración 62, Estadística mensual - mes más exitoso _____ 120
Ilustración 63, Grafico en barras uso diario - mes más exitoso _____ 121
Ilustración 64, Grafica pastel - Accesos por país - mes menos exitoso _____ 121

Ilustración 65, Acceso de los 4 primeros países - mes menos exitoso _____ 122
Ilustración 66, Estadística mensual - mes menos exitoso _____ 122
Ilustración 67, Grafico en barras uso diario - mes menos exitoso _____ 123
Ilustración 68, Modulo Bienvenidos - Parte 1 de 2 _____ 125
Ilustración 69, Modulo Bienvenidos - Parte 2 de 2 _____ 126
Ilustración 70, Pagina Tipo de Registro _____ 127
Ilustración 71, Modulo de Registro _____ 127
Ilustración 72, Modulo de logeo _____ 128
Ilustración 73, Pagina Bienvenido Cabecera _____ 128
Ilustración 74, Modulo de Perfil _____ 129
Ilustración 75, Modulo Edición de Perfil _____ 130
Ilustración 76, Modulo Búsqueda Cerca de ti _____ 130
Ilustración 77, Modulo Búsqueda General, _____ 131
Ilustración 78, Modulo Referencia de Ofertante _____ 132
Ilustración 79, Sección Solicite Pendientes _____ 132
Ilustración 80, Sección Me Solicitan Pendientes _____ 133
Ilustración 81, Sección Pendientes Contacto _____ 134
Ilustración 82, Sección Destacados _____ 134
Ilustración 83, Sección Footer Eliminar cuenta permanentemente _____ 135
Ilustración 84, Modulo Administración de Usuarios _____ 135
Ilustración 85, Modulo modificar cuenta por Administración de Usuarios _____ 136
Ilustración 86, Modulo insertar nuevo usuario por Administración de Usuarios 136

ÍNDICE DE TABLAS

Tabla 1, Baja de Usuario Administrador, 33
Tabla 2, Alta de usuario por Usuario Administrador 34
Tabla 3, Edición de Cuanta por Usuario Administrador 35
Tabla 4, Alta de Usuario ofertantes 40
Tabla 5, Baja de Usuario ofertante 41
Tabla 6, Alta de Usuario no ofertantes 42
Tabla 7, Baja de Usuario no ofertante 43
Tabla 8, Reporte de usuario 44
Tabla 9, Búsqueda de usuario Ofertante 45
Tabla 10, Tabla Carreras 55
Tabla 11, Tabla Especialidades 55
Tabla 12, Tabla Provincias 56
Tabla 13, Tabla Ciudades, 56
Tabla 14, Tabla Usuarios 57
Tabla 15, Tabla Ofertantes 58
Tabla 16, Tabla No ofertantes, 59
Tabla 17, Tabla Contratos 60
Tabla 18, Comparativa de Framework 74

INTRODUCCIÓN

La gran cantidad de información que se pone a disposición del público en la red, permite a los usuarios que la generan o proporcionan sumar a su audiencia *offline* una nueva audiencia *online*. El desarrollo de una plataforma *online* para promoción de servicios personales y profesionales, permitirá sumar nuevos miembros provenientes de la red al público general obtenido por medios convencionales, que es lo que precisa promover y facilitar el proyecto a su término.

Mencionado esto se puede establecer que el problema a tratar radica en la dificultad y demora para ubicar y/o contactar por parte del público general con servicios requeridos ya sean profesionales o personales. Puesto que, aunque se pueda encontrar algunos sitios *web* que pretendan ayudar a encontrarlos, estos no proporcionan información de forma ordenada y ágil, lo que en su gran mayoría genera más problemas por la pérdida de tiempo, confusión al contratar un servicio errado; o que no cumpla con los requerimientos del trabajo, lo que obliga a la búsqueda de otro y genera más costos por pérdida de tiempo y gastos que se pudieron evitar con la información correcta.

Al analizar la factibilidad del proyecto se realizaron algunas revisiones o comparaciones con sitios y aplicaciones que pretenden cumplir tareas similares a las que el proyecto realizará, como base de comparación se tomó a la página *web FreeLancer* la cual presenta varias características similares entre las cuales pueden ser mencionadas; la división de usuarios entre contratante y trabajador o la sub-clasificación de los usuarios clasificados como trabajadores pero que a su vez resultan diferentes, se enfoca solo en servicios profesionales; así como también el manejo de la información a modo de anuncios publicitarios para solicitar trabajo o un trabajador, diferencia claramente el proyecto y la plataforma mencionada anteriormente tanto en estructura como en funcionamiento, se llegó a considerar esta página como un sitio *web* de solicitud de empleo u empleados.

La comparación de las funciones o capacidad con una de las aplicaciones más usadas en la actualidad como lo es OLX, se muestran similitudes con el proyecto en desarrollo; la aplicación y su página *web* utilizan procesos apegados a una plataforma de anuncios que cuenta también con una sección dirigida a una clase

de usuarios que incluyen el público objetivo del proyecto de titulación, los cuales son personas que ofrecen servicios personales o profesionales, en la cual al revisar el funcionamiento de aplicación usada para la comparación (OLX) se encuentran diferencias radicales pues la aplicación mencionada no determina qué tipo de servicios se están ofertando o si es un servicio o una solicitud, lo que demuestra que no contempla procesos enteramente relacionados al proyecto.

Varios sistemas *online* que realizan tareas similares a las que podría realizar el proyecto de investigación y desarrollo existen en la red, ninguno realiza o implementa tareas como geolocalización o notificaciones vía SMS que se proponen desarrollar para el proyecto de investigación,

Preguntas Básicas

- **¿Cómo se origina el problema descrito?**

Falta de información para contactar y localizar un servicio.

- **¿Cuándo se origina el problema descrito?**

Necesidad de contratar y/o localizar un servicio para realizar un trabajo específico.

- **¿Cómo solucionará el proyecto propuesto el problema descrito?**

La realización del presente proyecto a realizar solucionará los inconvenientes a los usuarios permitiendo tanto ofertar sus servicios de forma ágil y ordenada como encontrar uno requerido de forma rápida y sin mayores confusiones, y se presenta la información de los mejores y más cercanos ofertantes, así proporciona a los usuarios la opción de saldar sus necesidades y evitar problemas de tiempo y localización.

Objetivos

Objetivo General

Desarrollar una plataforma *online* para promoción de servicios personales y profesionales.

Objetivos Específicos

- Fundamentar teórica y metodológicamente sobre el desarrollo de la plataforma *online*.
- Aplicar en la plataforma un diseño adaptativo o *Responsive Design.*
- Permitir la geo-localizar a los usuarios proveedores de los servicios.
- Generar un sistema de calificación de los servicios o *Ranked* determinado por los usuarios que requieran los servicios.

Delimitación del Proyecto

Funcional

- La plataforma *online* será capaz de adaptarse a cualquier dispositivo móvil al contar con *responsive design*.
- La plataforma *online* almacenará tres tipos de usuarios con sus respectivos datos los cuales se determinarán por el ambiente en el que se desempeñen, estos serán los usuarios ofertantes, no ofertantes y usuario administrador este último será el encargado del control de los usuarios no administradores.
- La plataforma *online* mostrará la localización de los usuarios ofertantes con respecto a los usuarios no ofertantes.
- La plataforma *online* permitirá a los usuarios no ofertantes calificar al usuario ofertantes del servicio requerido.
- Los usuarios ofertantes no serán personas jurídicas, pues la plataforma está pensada y orientada a las características de servicios individuales; y, ayudan a estos usuarios a promocionar sus servicios, pues al tomar en cuenta, que la gran mayoría de personas jurídicas ya cuentan con un sitio *web* de promoción propio se decidió centrarse en el apoyo a los servicios individuales.
- La calificación de un servicio no podrá ser realizada por usuarios ofertantes.

- Como menciona [1] el proyecto cumplirá con las características que definen un sitio *web* o plataforma *online*, se evita así la confusión con una página *web* entre las cuales se ponen de ejemplo que, un sitio *web* o plataforma está formado por varias páginas *web* pues una página *web* es la unidad básica dentro de la *world wide web,* así como, la plataforma requiere de un *webmaster* o administrador mientras que la página *web* no lo necesita.

Temporal

El proyecto de titulación fue desarrollado para el periodo del mes de octubre 2016 al mes de junio del 2017.

Justificación

El proyecto de titulación proporciona una plataforma *online* que permite localizar y contactar con varios servicios desde un mismo entorno de forma ordenada y precisa.

El público objetivo del proyecto serán los proveedores de servicios personales y profesionales; aclara que según la definición de profesional del estado ecuatoriano se considera a una persona proveedor de un servicio profesional una vez cuente con su título profesional registrado en el Senescyt, mientras esto no ocurra se considera como un servicio personal; los cuales serán denominados usuarios ofertantes, los mismos que al cumplir con determinados requisitos (Ej. Ruc y código de registro Senescyt) serán divididos a sus respectivas categorías (profesionales o personales) y subcategorías, de igual modo el público que realice la búsqueda de los servicios son clasificados como usuarios no ofertantes los cuales tienen que registrarse para acceder a la totalidad de la información disponible de los ofertantes.

La plataforma *online* contará con un diseño responsivo (*responsive design*) permitiendo que pueda ser usada cómodamente desde cualquier dispositivo móvil que cuente con conexión a internet; así mismo contara con la opción de calificar y comentar los servicios que se han contratado, esta calificación se realiza únicamente por usuarios registrados en la plataforma, esto pretende brindar seguridad al contratar un servicio, pues la calificación y los comentarios proporcionan nociones reales sobre la calidad del servicio y sus capacidades; así

como asegurar que otros usuarios registrados como ofertantes de la misma rama de servicios generen información falsa al calificar o comentar de forma negativa a sus competidores.

El proyecto implementa un sistema de geo-localización de los servicios registrados proporciona la ubicación de estos con respecto a la posición en la que se encuentre el usuario que realiza la búsqueda en la plataforma, permite la identificación y locación de los usuarios ofertantes, esta función también ofrece confianza y seguridad a los público al proporcionar la posibilidad de realizar una solicitud de forma directa en el caso que sea necesario.

Se concluyó motivada por la comparativa de características, funcionalidad y orientación con algunos sitios *web* que promocionan servicios similares, que los sitios existentes actualmente están pensados en: realizar solicitudes y esperar respuesta a estas, la promoción de una rama de profesiones con algún tipo de relación entre ellas o publicación de servicios y productos no categorizados y fomenta así nuestro deseo de crear una plataforma dedicada con las características antes mencionadas.

Meta

Desarrollo de una plataforma *online* que permita la administración de los servicios personales y profesionales y contar con un mínimo de cien usuarios ofertantes registrados.

Esta meta fue propuesta por el directorio encargado de la aprobación del plan del proyecto de titulación.

CAPITULO I. ESTADO DEL ARTE Y LA PRÁCTICA

1.1. FUNDAMENTOS TEÓRICOS

1.1.1. Internet

Los inicios del internet se dieron a finales de los años sesenta, el supuesto más relevante de su origen es de base militar, el cual menciona fue desarrollado por y para el ejército de los Estados Unidos de Norteamérica. Con el paso del tiempo este evoluciona a un uso de nivel mundial desarrollándose aplicaciones con diversos fines entre los cuales destacan los fines comerciales, académicos, y sociales [2].

Con lo antes mencionado se puede aclarar que el internet es una combinación de hardware interconectado entre sí por cables o de forma digital a lo largo de todo el mundo entre los cuales se comparte información entre uno o varios puntos usando software, lenguajes, protocolos y programas que interpretan y permiten la visualización de la información compartida.

Actualmente el internet es una de las herramientas más relevantes en el mundo esto se puede evidenciar en la producción de dispositivos electrónicos modernos que incluyen la capacidad de acceder a internet [3].

1.1.2. Sitio *web*

Se define como un sitio *web* al conjunto organizado y correctamente relacionado de páginas *web* que están integrados en un determinado dominio en Internet el cual se puede visualizar a través de la *World Wide Web* (www) mediante navegadores *web* tales como *Chrome*, *Firefox*, *Edge*, *Opera* entre otros, las páginas *web* que conforman los sitios *web* están formados por archivos de formatos html, php, css, entre otros [4].

1.1.3. Diferencia entre sitio *web* y pagina *web*

Es muy común el uso erróneo de los términos página *web* para hacer referencia a un sitio *web*, una página *web* es una parte de un sitio *web* y es un archivo individual y único con un nombre definido a su vez un sitio *web* es el conjunto de 1 o más archivos llamados páginas *web*.

Un sitio *web* se puede ejemplificar como un libro, en el cual el título del libro sería el nombre del dominio del sitio *web* y cada capítulo que conforma el libro sería una página *web* que conforma el sitio *web*, así mismo cada capítulo tiene un nombre único de igual forma cada página *web* tiene asignado un nombre único.

La página *web* se compara con el capítulo y no con una página del libro porque una el contenido de una página *web* consta de un inicio y un final detallados, tal como un capítulo de un libro, no así una sola página de un libro no tiene un inicio o final detallados como tal [5].

1.2. DESCRIPCÓN DE INGENIERÍA *WEB*, ARQUITECTURA *WEB* Y APLICACIONES *WEB*

1.2.1. Ingeniería *Web* (I*Web*)

Actualmente los sistemas y aplicaciones basados en *Web* brindan una gran cantidad de contenido y funcionalidades a buena parte de los usuarios finales, la ingeniería *Web* es un proceso a través del cual se generan o desarrollan *WebApps* (son sitios *web* que han sido desarrollados de modo que estén optimizados con un diseño y organización de la información para ser vista desde dispositivos móviles) que consiguen ocupar los mejores estándares de desarrollo *web*. No es exactamente como la Ingeniería del *Software*, pero usa muchos conceptos y principios fundamentales [6].

Las *WebApps* se han tornado un estándar en los últimos años, pues se han integrado de forma sólida en los negocios de empresas grandes, medianas y pequeñas. Está en la principal razón de la importancia de construir sistemas fiables basados en los principios de la ingeniería del *software*.

La Ingeniería *Web* comienza con la investigación de requisitos, continúa con un análisis detallado y diseño o prototipo, pasando a una implementación con tecnologías y herramientas específicas para la *Web*. Finalmente se revisa el resultado con los clientes y se refina, estableciendo un control de versiones y configuraciones que asegure una calidad de la aplicación y un posible soporte a largo plazo que fije su evolución.

1.2.2. Arquitectura *Web*

La arquitectura *Web* permite a múltiples clientes o usuarios establecer una comunicación a través de Internet con un servidor *Web*.

Los clientes mediante un navegador emiten peticiones vía HTTP a un servidor *Web* que responde a cada una de las solicitudes.

El servidor aloja una serie de aplicaciones *Web* que ofrecen diferentes servicios a los usuarios conectados. Es en el servidor *Web* donde se centraliza la carga de trabajo. En la mayoría de los casos, el navegador suele ser un mero presentador de información (modelo de cliente ligero), y no lleva a cabo ningún procesamiento relacionado con la lógica de negocio.

En la arquitectura *Web* el modelo de comunicación entre cliente y servidor se establece mediante una conexión a Internet o intranet. El intercambio de información desde el navegador al Servidor se efectúa mediante el uso del protocolo *Hipertext Transfer Protocol* (HTTP) protocolo para el intercambio de información en *Web*. El navegador *Web* interpreta las páginas dinámicas generadas por el servidor *Web* [7].

Se toma como referencia el texto anterior, se expresa con exactitud el concepto de arquitectura *Web* y el funcionamiento del sistema, establece de modo simple y detallado las acciones y el funcionamiento que tiene el sistema referido y su capacidad de proporcionar en la actualidad un método o forma más rápida de trabajo o comunicación de los usuarios de estos servicios con el mundo, a su vez la capacidad de almacenamiento de información relacionadas con las *WebApps*.

1.2.3. Aplicaciones *web*

“Una aplicación *Web* es una aplicación informática que se ejecuta en un entorno *Web*.” Las aplicaciones *web* son un tipo de software que se codifica en un lenguaje soportado por los denominados navegadores *web* y cuya ejecución se lleva a cabo por el navegador en Internet o de una intranet (de ahí que reciban el nombre de *App web*) [8].

Otra definición que se puede dar para una aplicación *web* es la siguiente:

Son aquellas herramientas que los usuarios utilizan accediendo a un servidor *web* por medio de *Internet* o de una intranet mediante un navegador que las ejecuta.

Que las aplicaciones *web* sean ejecutadas por medio de un navegador *web* en una red significa que los datos o los archivos en los que trabajas se han procesado y almacenado. Por este motivo, este tipo de aplicaciones por lo general, no necesitan ser instaladas en el computador o el celular.

Una página *Web* puede contener elementos que permitan comunicarse en forma activa entre los usuarios y la información, haciendo que se pueda acceder a los datos de forma interactiva, el sitio *web* se encarga de responder a cada una de las acciones que éste establezca (por ejemplo, acceder a gestores de bases de datos de todo tipo, publicar e interactuar con los contenidos, rellenar y enviar formularios, participar en juegos, entre otros).

Las aplicaciones *web* están íntimamente relacionadas con el almacenamiento de datos, toda la información se guarda de forma permanente en servidores, los cuales además de alojar dicha información, la envían a nuestros dispositivos móviles o equipos informáticos en cada momento que sea requerida, y efectua copias temporales de estos envíos dentro de los equipos y dispositivos que utilicemos [9].

Las aplicaciones *web* son muy populares debido a:

- La practicidad que ofrecen los navegadores *web* como clientes ligeros. - Un cliente ligero (*Thin Client*) es un sistema que trabaja en una arquitectura de red cliente-servidor en la cual existe muy poca o ninguna lógica del programa, por lo que depende principalmente del servidor central para las tareas de procesamiento.

- La independencia del sistema operativo que se use en el computador o dispositivo móvil.
- La facilidad para actualizar y mantener aplicaciones *web* sin necesidad de tener que distribuir el *software* o que se tengan que instalar los usuarios.
- El libre acceso en cualquier momento, lugar o dispositivo, sólo con tener conexión a Internet y los datos de acceso (nombre de usuario y contraseña).

1.3. VENTAJAS Y DESVENTAJAS DE LAS APLICACIONES *WEB*

1.3.1. Ventajas de las aplicaciones *Web*

Compatibilidad multiplataforma: pueden ser usadas desde cualquier Sistema Operativo siempre que se cuente con un programa buscador actualizado.

Actualizaciones inmediatas: Las actualizaciones son constantes y hechas por el desarrollador de la *WebApps* sin necesidad de intervención del usuario.

Acceso inmediato: Al ser aplicaciones en línea no necesitan ser instaladas o descargadas, su carga es inmediata (acorde a su conexión de internet) y se pueden usar rápidamente.

Consumo bajo de recursos: Al ser desarrolladas de forma correcta el uso de recursos del equipo en el cual se está haciendo uso de esta *WebApps* será muy bajo o casi nulo.

Reducción costes: Se reducen costos en cuestión de mantenimiento de equipos, estos son realizados por los proveedores del servicio de hosting, así como se evita el costo de establecer un servicio de host propio.

Alta disponibilidad de servicio y datos: gracias a la conectividad que proporciona Internet se permite el acceso a la aplicación desde cualquier punto. Esto facilita la movilidad y el teletrabajo.

Portabilidad: al estar en línea su ejecución es posible en diversos lugares siempre y cuando se cumplan las condiciones para la ejecución de un sistema *online*.

1.3.2. Desventajas de las aplicaciones *Web*

Conexión a Internet: se necesita obligatoriamente conexión a internet.

Comunicación no segura: La comunicación puede estar en Riesgo por ataques de intercepción de información.

Seguridad en el servidor: La *WebApps* dependerá en gran medida de las seguridades con las que cuente el servido que le da alojamiento.

Dependencia a plugins: algunas funciones pueden depender de complementos que no estén activos en el navegador del usuario.

Espacio reducido de almacenamiento: El espacio de almacenamiento dependerá del servicio que se ha contratado al proveedor de hosting.

1.4. TIPOS DE APPS *WEB*, TECNOLOGÍAS, ENTORNOS DE EJECUCIÓN E IMPLEMENTACIÓN

1.4.1. Tipos de Apps *web* que se pueden desarrollar

A continuación, se presentan algunos ejemplos de aplicaciones *web* que se pueden llegar a desarrollar, todos de estos tipos forman parte de cotidianidad y forman parte de nuestro entorno tanto virtuales como reales [10].

***Web mail*:** son sistemas de acceso a el correo electrónico que admiten acceder a tus correos por medio de un navegador *web*, sin tener que descargar los correos en el ordenador. Para ello utilizan clientes del tipo Gmail, Outlook, etc.

***Wikis*:** son sitios y aplicaciones *web* cuyas páginas y contenidos pueden ser editados desde el navegador y donde los usuarios pueden crear, modificar o eliminar contenidos que por lo general comparten.

***Weblogs*:** son sitios y aplicaciones cuyas páginas y contenidos son de fácil actualización, de tal manera que permiten a sus autores publicar contenidos (textos, imágenes y otros archivos) con presionar un solo botón, suelen contar con un editor y herramientas para tal efecto.

***Tienda virtual o comercio electrónico*:** estos son tipos de comercio que usan como medio principal un sitio *web* y/o una aplicación conectada a internet desde la que los usuarios y/o clientes pueden realizar transacciones o sus compras.

Portal web app: Con portal hacemos referencia a aplicaciones las cuales cuentan una página principal que permite el acceder a una serie de apartados, categorías o secciones. El portal *web* puede contar con todo tipo de información: foros, chats, correo electrónico, un buscador, zona de acceso con registro, contenido más reciente, etc.

En el caso de la plataforma planteada en este proyecto, se considera dentro de la última categoría siendo un portal *web* destinado a proporcionar información de usuarios que deseen ofertar o contratar un servicio.

1.4.2. Tecnologías, entornos de ejecución e implementación

HTML:

Es un lenguaje diseñado e implementado por Tim Berners-Lee a finales de los años ochenta, y fue presentado a la comunidad en el año de 1990. El lenguaje está basado originalmente en el *HyperText* y SGML, creando una clase de *HyperText global*, utilizando la nomenclatura DNS y con un protocolo propio, para la recuperación de archivos a través de enlaces *HyperText* (HTTP) [11].

CSS:

Por su nombre en español Hojas de estilo en cascada, es un lenguaje que permite añadir aspectos gráficos a los documentos HTML. Sin embargo, *Tim Berners Lee* en 1990 separó la estructura de los documentos HTML de su representación gráfica, no fue publicado por *Håkon Wium Lie* y *Bert Bos* hasta el mes de abril de 1995 en la conferencia WWW [12].

PHP:

Un lenguaje de scripting que se ejecuta en servidores *web*, es el sucesor de un producto llamado PHP/FI. Creado en 1994 por Rasmus Lerdorf, la primera versión fue un conjunto simple de archivos binarios CGI escritos en C. Con el tiempo llamó al conjunto de scripts "*Personal Home Page Tools*", más conocido como "*PHP Tools*". En Julio de 2004 fue lanzado PHP 5, con un núcleo *Zend Engine* 2.0 que contiene un nuevo modelo de objetos y docenas de nuevas opciones [13].

SQL:

En la década de los setenta, después de que E.F. Codd presentara su propuesta sobre el Modelo Relacional, aparecieron varios productos comerciales que seguían esta propuesta. En la década de los 80 el organismo ANSI publicó la primera versión estándar del SQL (*Structured Query Language*). A partir de ese momento, este organismo publicó varias versiones que mejoraban las deficiencias encontradas en las versiones previas e introducían las mejoras que las investigaciones en el campo de las bases de datos iban descubriendo [14].

JavaScript:

Es el lenguaje de programación interpretado que dio surgimiento al estándar *ECMAScript* usualmente utilizado como código embebido en documentos HTML como archivos .js dentro del código HTML este lenguaje es ejecutado en la mayoría de casos en el navegador del usuario no en el servidor. Este lenguaje se utiliza para añadir interacción con el usuario a los documentos HTML estáticos. Fue creado en mayo de 1995 por Brendan Eich quien trabajaba en Netscape [15].

Google Maps Platform

Esta es una de las plataformas para desarrolladores de *Google* disponible en *Google developers* la cual cuenta con varias herramientas de desarrollo e implementación como son *Geocoding API, Geolocation API, Maps JavaScript API* las cuales fueron adquiridas para la creación de esta aplicación, al momento de adquirir dichas herramientas se proporciona al desarrollador una *API KEY* de Google que dependiendo de las aplicaciones deseadas se pueden restringir o habilitar para el uso exclusivo del proyecto [16].

1.4.3. Sistema de Notificaciones

La necesidad de notificación a los usuarios de los procesos realizados por y en la plataforma hizo necesario que se notifique a los usuarios a través de mensajes enviado directamente su email registrado en la plataforma, permitiendo a los usuarios ser contactados.

Existen varias compañías a nivel mundial que ofrecen el servicio de notificación sin embargo es necesario el conocimiento básico de programación PHP para poder implementar el servicio.

1.5. HERRAMIENTAS DE EJECUCION Y DESARROLLO

Las herramientas de ejecución y desarrollo deben cumplir con los requerimientos mínimos especificados continuación.

1.5.1. Plataforma de Ejecución

- Sistema Operativo: Windows Server 2012
- Sistema Gestor de Base de Datos: MySQL versión 10.2.17
- Lenguaje de Desarrollo en Servidor: HTML5, PHP 5.1.0 y JavaScript 1.8.5
- Servidor *Web*: Apache

1.5.2. Herramientas de Redacción y Desarrollo

- Sistema Operativo: Windows 7 Pro
- Herramienta FTP: FileZilla_3.22.1 64bits
- Herramienta de codificación: Subtext line 3.1.1
- Herramienta de composición de documentos: Office 365 (compatible con todas las versiones de Microsoft office predecesoras)
- Herramienta de gestión de diagramas UML: Herramienta *online* https://www.lucidchart.com y StarUML 3.1.0
- Herramienta de diseño gráfico y retoque: Gimp 2.8.18

CAPITULO II. DISEÑO METODOLOGICO PARA EL DESARROLLO DE UNA PLATAFORMA ON LINE

2.1. MÉTODOS

Método: Análisis de Documentos

Este método de investigación consiste en el escrutinio detallado de documentos donde los analistas de sistemas y diseñadores tratan de encontrar la información que permita dar inicio a nuevas investigaciones. Entre los documentos más comunes se puede encontrar la historia y documentación de entidades, estado económico, tendencias socio-culturales entre otras [17]. Se revisó la documentación de plataformas que manejan ideas similares a esta propuesta, de las cuales se deben mencionar la documentación de [18] en la cual se expresa que el objetivo de dicha plataforma es compilar propuestas para la realización de proyectos, a diferencia de la idea presente para el proyecto.

Del mismo modo se revisó la documentación proporcionada por [19] revela que a pesar de ser una plataforma dirigida principalmente al manejo de anuncio de productos cuenta con una sección dirigía a la promoción de servicios, sin embargo, esta no cuenta con las características que se exponen más adelante en este documento (orientación específica de servicios individuales de los usuarios, geolocalización, notificación instantánea. etc.) Por tanto, el análisis de la documentación relevante avala el desarrollo del proyecto tomando en cuenta que, aunque comparte estrategias similares a plataformas existentes.

2.1.1. Método OOHDM y sus etapas para la aplicación

La método *Object Oriented Hipermedia Design Method* (OOHDM), diseñada y explicada en [20], donde los autores mencionan que las metodologías tradicionales de ingeniería de software así como las dirigidas a sistemas de desarrollo de información no contienen una buena abstracción capaz de facilitar la tarea del desarrollo *web*; destacando también que el método OOHDM está orientada a objetos reconoce el uso de conceptos como los expuestos en [21]. Esta metodología cuenta con cinco etapas las cuales se aplicaron en el desarrollo del proyecto de la siguiente manera.

2.1.2. Primera Etapa. Obtención de Requerimiento

En esta etapa se usarán herramientas como el diagrama de casos de uso, para determinar las acciones que se desea realice la plataforma *online* de acuerdo a cada escenario presentado. El primer paso necesario identifica a los actores y que tareas serán llevadas a cabo por ellos. El siguiente paso es determinar los escenarios de cada una de las tareas que desarrollaran los diferentes tipos de actor. Los casos de uso generados en los pasos anteriores pasaran a ser representados con Diagramas de Interacción de Usuario, los mismos que permiten representar gráficamente y de forma más precisa la interacción entre usuarios y el sistema mientras se está ejecutando una tarea. Este tipo de diagramas capturan los requisitos de la plataforma independientemente de una implementación [22].

Esta etapa es una de las más importantes, aquí se realiza la recopilación de los datos, para lo cual se realizarán preguntas como las siguientes:

“¿Cuáles son los tópicos principales que serán atendidos?”

“¿Cómo los tópicos están relacionados entre sí?”

“¿Qué categoría de usuarios serán atendidos?”

“¿Cuáles son las tareas principales que serán abordadas?”

“¿Qué tareas corresponden a qué categoría de usuarios?”

“¿Los recursos disponibles son competitivos con la información levantada?”

Con las respuestas a las preguntas anteriores, se pueden recopilar una base necesaria para la construcción de una plataforma o aplicación *web* exitosa, sin embargo, mientras más se profundice en la reunión de información y datos, mayor será la probabilidad de satisfacción de los usuarios [23].

2.1.3. Segunda Etapa. Modelo Conceptual

En esta etapa, se construyó la síntesis gráfica que sirvió para representar los objetos incluyendo en dicha síntesis, las relaciones entre ellos y el dominio de destino de la plataforma o aplicación *web*, estos elementos no son modificados durante el funcionamiento de las acciones representadas en la gráfica [24]. Hay

que resaltar que si se desea que la base de información pueda modificarse de forma dinámica o que la aplicación realice funciones complejas como consultas que tenga relación directa con los objetos, será necesario aplicar un modelo orientado a objetos mejor elaborado y desarrollado.

OOHDM es una metodología basada en clases, relaciones y subsistemas. Las clases se describen con modelos orientados a objetos representando diferentes perspectivas de una misma entidad usando notación similar a *Unified Modeling Language* (UML) [25] que nos permite definir el diseño de decisiones *forward* y *backward.* Diferentes procesos de modelado y diseño de objetos son propuestos y discutidos por metodologías orientadas a objetos y bibliografía en el campo.

En OOHDM la síntesis gráfica de clase consiste en un conjunto de clases conectadas por relaciones en donde los objetos son instancias de las clases, por lo tanto, cuando una relación se cumple entre las clases se abstrae las relaciones objeto a objeto correspondientes. Estas clases podrían ser utilizadas más adelante durante el diseño navegacional para proceder a otros nodos y relaciones y estas a su vez pueden ser utilizadas para obtener enlaces.

En esta síntesis gráfica de clases, se ha enfatizado las relaciones sobre el comportamiento de clase por simplicidad. Los atributos de las mismas se escriben y representan los objetos específicos, es decir como lo serian el título de la historia, el nombre del autor, etc. Cada tipo posible es llamado una perspectiva, similar a *A Model-Based Approach to Hypertext Application Design* (HDM) [26], cuando múltiples perspectivas existen, se utiliza la notación p1, p2 (perspectiva1, perspectiva2, etc) y se puede añadir un signo más (p1+) junto a una perspectiva para marcarla como perspectiva por defecto. Solo la perspectiva por defecto debe estar presente en todos los casos, mientras que las otras perspectivas pueden o no ser aplicadas.

Durante la etapa de diseño navegacional y diseño de interfaz abstracta se refina estas especificaciones definiendo qué perspectiva se muestra, cuándo y cómo. Claramente, cuando implementamos clases en el modelo conceptual, necesitamos especificar una variable de instancia para cada perspectiva [27].

Decidir si expresar una relación como un atributo, un método o una combinación de ambos es un problema de diseño que ha sido ampliamente discutido en la comunidad de desarrollo orientado a objetos. Actualmente los enfoques de modelado orientados a objetos utilizan las relaciones como entidades Ciudadanos de primera clase (*First Class Citizen:* lo que significa que es una entidad que puede soportar las operaciones generalmente disponibles para otras entidades en su totalidad), relegando así la discusión a la etapa de diseño e implementación. Como alternativa, los enfoques conductuales como el diseño dirigido por *Wirfs-Brock* [25] favorecen el uso de diagramas de colaboración en lugar del uso de relaciones al estilo de UML.

Por otra parte, en la aplicación del diseño *web* o hipermedia, las relaciones son un punto de gran relevancia dentro del dominio de la aplicación y no deben ser ocultados como atributos de clase; lo que da como resultado que una relación debe detallar que un atributo figure como una entidad conceptual compleja, que se explorará en la plataforma o aplicación *web* final. Se recalca que al implementar aplicaciones *web* utilizando una arquitectura orientada a objetos la clase de origen tiene la responsabilidad de dejar accesos a las relaciones implementadas como un método.

En el enfoque de diseño, se ofrecen tres opciones de abstracción para tratar con la complejidad: Agregación, Generalización o Especialización y Sub-Sistemas. Agregación se utiliza para describir clases complejas, mientras, Generalización o Especialización es útil para construir jerarquías de clases, y usar la herencia como un mecanismo de uso compartido. Los sub-sistemas son una forma de empaquetado para abstraer modelos enteros del dominio.

Parte de las relaciones se describen al utilizar *relaciones de agregación*. Los agregados en una definición de clase son similares a los componentes en HDM [26]. Existen relaciones implícitas entre un objeto complejo y sus partes, y entre las partes mismas, estos se llaman enlaces estructurales en HDM.

Dado que la agregación es una relación estructural importante, se sugiere descomponer un objeto en partes cada vez que estas partes se supone que se

exploran en la plataforma. La heurística existente en el dominio de modelado orientado a objetos puede usarse para identificar estructuras de agregación.

El uso de un modelo orientado a objetos conductuales para señalar varias características de una plataforma o aplicación *web* permite enumerar una variedad de actividades informáticas, como consultas dinámicas a una base de objetos, modificaciones de objetos, búsquedas, entre otros. El Modelo conceptual depende de las características de la aplicación que se especificaron o se quiere que estén presentes en la aplicación. En una gran cantidad de aplicaciones *web* especialmente en las cuales se efectúan funcionalidades de navegación sencilla, el comportamiento de clase que realicen acciones adicionales aparte de la función del enlace puede juzgarse como inutilices o innecesario. En este caso particular, el comportamiento se considera igual a acceder a una base de datos que navega en las relaciones.

Por el contrario, en aplicaciones en las que la base de datos puede cambiar de forma dinámica como la intervención del usuario en la plataforma o aplicación *web*, para trabajar correctamente con dichas interacciones se implementan métodos de comportamiento en clases conceptuales. En el proceso de diseño Navegacional y de Interfaz, se especifican las formas en que los objetos de interfaz actúan en los modelos de navegación y en los conceptuales.

Dado que OOHDM es un método de diseño y no un marco de implementación, se usó métodos para obtener el valor de los atributos de un objeto. El uso de términos de metodologías de desarrollo orientado a objetos, se expresa que en el Modelado Conceptual las características de los Modelos de Análisis y Diseño deben ser detalladas. Cuando la aplicación se ejecuta en la parte superior de un entorno que soporta objetos, las clases del modelado conceptual se implementarán de forma directa según se definan, especificando múltiples perspectivas como atributos separados.

2.1.4. Tercera Etapa. Estructura Navegacional

El método OOHMD cuenta con varias clases predefinidas que ayudarán a definir la estructura que tendrá la plataforma *online* basándose en las etapas anteriormente

mencionadas al mismo tiempo que otorgará flexibilidad al momento de estructurar la plataforma y así esta sea adaptada a las necesidades de los usuarios, además de permitirnos el uso de reglas SEO como lo sugiere [7] y [8].

En el método OOHDM, la correcta estructura de navegación se considera un punto crítico en el desarrollo de la aplicación *web*. Esta etapa se desarrolla observando de forma detallada la etapa anterior (modelado conceptual), de este modo nos permite crear diferentes esquemas de navegación dependiendo de cada perfil de usuario. La estructura navegacional nos permite observar un esquema individual del modelado conceptual. En la elaboración de la estructura navegacional se toman en cuenta diferentes aspectos como:

- ¿Cómo se realiza la navegación cuando el usuario va de un objeto a otro?

Como afecta la navegación al objeto fuente y objetivo, y como afecta está a otros objetos y objetivos de la estructura navegacional [20].

En OOHDM, la estructura de navegación de una aplicación se puede dar por un esquema de clases de navegación específicas que definen un contexto de navegación como un conjunto de nodos, enlaces, clases de contexto y otros contextos de navegación. Puede definirse intencional o extensamente, ya sea definiendo una propiedad que poseen todos los nodos y enlaces en el contexto, o enumerando sus miembros. La definición de un contexto también incluye el orden de sus elementos, y la existencia o no de estructuras de acceso asociadas.

Nodos

Los denominados nodos contienen información básica de las aplicaciones hipermedia, que pueden definirse como reflejos de las clases definidas en el diseño conceptual con un desarrollo encaminado a objetos usando lenguajes predefinidos e intuitivos, admitiendo que sean definidos a través de la combinación de atributos de diferentes clases relacionadas en el diseño conceptual.

Los nodos principalmente contienen atributos de tipo básico en donde se pueden hallar imágenes, sonidos y enlaces.

Enlaces

Reflejan la relación de navegación que puede explorar el usuario. Se sabe que para un mismo esquema conceptual puede haber diferentes esquemas navegacionales y que los enlaces son imprescindibles para poder crear esas diferentes vistas.

OOHDM hace una definición de la clase enlace que contiene un único atributo que almacenaría la clase a la que se navega por ese enlace. Las clases enlaces especifican los atributos de enlaces y estos a su vez representan los enlaces entre clases nodos o incluso entre otros enlaces. En cualquier caso, el enlace actuara como un objeto intermedio en un proceso de navegación o como un puente de conexión entre dos nodos.

Estructuras de Acceso

Las estructuras de acceso actúan como índices o diccionarios que facilitan al usuario encontrar de forma rápida y eficiente la información deseada. Ejemplos de estas estructuras son los menús, los índices o las guías de ruta. Las estructuras de acceso igualmente se modelan como clases, compuestas por un conjunto de referencias a objetos que son accesibles desde ellas y una serie de criterios de clasificación de las mismas.

Contexto Navegacional

Para diseñar bien una aplicación hipermedia, hay que prever los pasos que el usuario puede seguir, únicamente así es como se puede evitar información redundante o que el usuario se pierda en la navegación. En OOHDM un contexto navegacional está compuesto por un conjunto de nodos, de enlaces, de clases de contexto y de otros contextos navegacionales. Estos son introducidos desde clases de navegación (enlaces, nodos o estructuras de acceso), pudiendo ser definidas por extensión o en forma implícita.

Clase de Contexto

Es otra clase especial que sirve para complementar la definición de una clase de navegación. Por ejemplo, sirve para indicar qué información está accesible desde un enlace y desde dónde se puede llegar a él [28].

Los conceptos ante descritos en los subtemas del literal del 2.1.4 se han tomado textualmente, pues describen de una forma sólida y especifica por qué son elementos fundamentales de esta etapa y cuál es su uso o función dentro del desarrollo de esta etapa.

2.1.5. Cuarta Etapa. Interfaz Abstracta

En esta etapa se definirá que objetos percibirá el usuario; y, el orden en que estos aparecerán en la interfaz según el avance en la plataforma, así también se definirá la sincronización de los objetos y el interfaz de transformación, lo que permite utilizar conceptos de diseño *web* como los establecidos en [9].

Una vez definida la estructura de navegación de la plataforma o aplicación *web*, especificaremos sus aspectos de interfaz, lo que significa definir como aparecerán los diversos objetos de navegación, los objetos de la interfaz que permitirán la navegación y otras funciones de la plataforma o aplicación *web* y qué cambios sufrirá la interfaz.

Una delimitación clara entre diseño navegacional y diseño de interfaz abstracta, proporciona la capacidad de desarrollar diferentes interfaces para una misma estructura navegacional. Además, la delimitación de estas proporciona un mejor panorama de la estructura de la plataforma o aplicación *web* al permitir la definición de qué cambios en la interfaz son también cambios en la estructura navegacional, además de mostrar que cambios en la interfaz del usuario no afectan a la estructura navegacional y por lo tanto no requieren acceso a las funciones del servidor.

Pese a haber argumentado que el aspecto de la interfaz de usuario de la plataforma *online* es un componente crítico, las metodologías generales desatienden este aspecto. Designan las especificaciones de la interfaz a herramientas dependientes de la implementación, y por lo tanto los detalles de cómo se desarrolló el diseño de

esta etapa no son documentadas adecuadamente o simplemente no son documentadas.

Desde otro punto, el desarrollo y la implementación de la interfaz de la plataforma *online* suele hacerse por medio de editores HTML, PHP y CSS, y sin la adecuada descripción en el desarrollo de la interfaz, muchos aspectos críticos pueden ser ignorados.

El método OOHDM, utiliza un enfoque de diseño de vistas de datos abstractos para describir la interfaz de usuario de una plataforma o aplicación *web* [29]. *Abstract Data Views* (ADV) son objetos, los cuales tienen un estado y una interfaz, donde el usuario puede ejercer acciones sobre la plataforma *online* a través de eventos externos o acciones directas en la plataforma o aplicación *web*. Los ADV también son considerados representaciones de la interfaz y estado, pero no de la implementación de los mismos.

Estos pueden utilizarse para representar interacciones entre diferentes medios como pueden ser los usuarios, la red en la que funciona la plataforma o aplicación *web* o un dispositivo enlazado a esta.

Los mismos también son usados para representar la interfaz entre dos o más objetos de datos abstractos o *Abstract Data Objects* (ADO). Un ADO es un objeto que no admiten cambio o eventos externos que hayan sido provocados por usuarios [29]. Si se observa el comportamiento o utilidad de los ADV y ADO en la arquitectura de la plataforma o aplicación *web*, se puede decir que los ADV son observadores de ADO, por lo que la comunicación entre objetos de interfaz y la plataforma o aplicación *web* cumple con las normas definidas en el *Observer Design Pattern* o patrón de diseño observador [30].

Los ADV usados para el desarrollo del diseño de una aplicación *web* pueden ser muy similares a los objetos de interfaz, estos están formados por un conjunto de atributos y uno o varios objetos de interfaz anidados que especifican o determinan las propiedades de percepción de los mismos, así como también contienen un conjunto de eventos que puede asimilar como provocados o realizados por un determinado usuario. Varios ejemplos de eventos provocados o realizados por un

usuario son clic con el ratón, doble clic con el ratón, scroll del ratón hacia arriba o abajo, clic sostenido del ratón, entre otros.

Una de las mejores particularidades de los ADV es que su implementación en entornos de desarrollo orientados a objetos para aplicaciones *web* resulta rápidamente sencilla, así como también pueden interpretarse como documentos HTML, PHP y CSS con la misma facilidad.

Se definen como constantes los valores de los atributos dándoles cualidades únicas de apariencia como son la ubicación, tamaño, sonido y color. El uso de ADV en el modelado de la interfaz facilita el manejo de dichas características de forma abstracta, relegándolas a la fase de implementación. El comportamiento y la organización de la interfaz son detallados en ADV pero es en la fase de implementación en donde la apariencia física real de los atributos y los detalles especificados en los en estos es desarrollada.

También se pueden reservar variables las cuales se pueden usar como indicadores de cambios dentro del espacio que percibe el usuario, de este modo si se desea o necesita que un objeto sea percibido por el usuario es añadido a una de estas variables, así como los elementos que se desea no sean perceptibles son eliminados de dichas variables.

Se estipula que en el marco del método OOHDM los nodos e índices se consideran o actúan como ADO y la apariencia relacionada con cada uno de ellos se especifica en sus correspondientes ADV. Así también se aclara que el diseño de estos, toma diferentes mecanismos de abstracción y composición en su desarrollo; Al inicio pueden ser formados por agregación o composición de ADV de nivel inferior, lo que admite y proporciona la facilidad de construir interfaces de usuario con objetos percibidos anidados.

2.1.6. Quinta Etapa. Implementación

Una vez cumplidas las etapas anteriores se traslada todo lo realizado a un lenguaje de programación adecuado para cumplir con lo determinado en el proceso. En esta fase, el desarrollador implementará el diseño construido previamente con la fusión de todas las fases antes realizadas. Todas las fases anteriores fueron elaboradas

para que el resultado de cada una de ellas fuera independiente de la plataforma desarrollada.

Esta etapa del método trabaja no solo con las fases anteriormente desarrolladas si no también con el entorno en el que va a ejecutar finalmente la plataforma, tomando esto en cuenta hay que concentrarse en como el método OOHDM y todos los diseños desarrollados a través de cada una de las fases, pueden ser implementados finalmente en la red *World Wide Web* (WWW*)*, la realización este paso considera también que existen varias alternativas a través de las cuales se puede cumplir este objetivo [31], como por ejemplo el uso de *frameworks* que pueden ayudar a una implementación de la etapa de diseño más rápida y firme, o el desarrollo desde cero que permite conocer con precisión las partes más profundas de la etapa.

En este punto, las anteriores fases ya deben haber definido los elementos informáticos con los que trabajarán, evitando así problemas con el servidor de alojamiento; o funciones que no sean necesarias para que la plataforma trabaje correctamente. Siguiendo con el correcto desarrollo de la plataforma se debió identificar previamente los elementos que formarán parte de cada tipo de perfil de usuario y las acciones que cada uno de estos pueden realizar. Se tendrá bien definido el comportamiento de la interfaz con cada una de las acciones y su estructura de funcionamiento.

En el comienzo de la implementación, el desarrollador de la plataforma determina cómo será almacenada la información, tanto de las etapas de modelado conceptual y estructura navegacional, definiendo también como trabajará la interfaz de la plataforma y el comportamiento que tendrá la plataforma en si al usar los lenguajes de programación como PHP y HTML; contemplando la posibilidad de necesitar extensiones o librerías que faciliten o mejoren el desarrollo de la aplicación, y de ser necesaria la asesoría de un profesional de diseño gráfico deberá documentarse adecuadamente su participación en el proyecto.

Para finalizar se resalta que el método OOHDM aplica los principios de la programación orientada a objetos sin embargo no necesita de un entorno de implementación del tipo mencionado anteriormente.

2.2. USO DE OOHDM Y SUS DIFERENTES MODELOS

La creación de un plan del sistema de información en la *web* usa los diseños y modelos desarrollados en la etapa cuatro: interfaz abstracta, sin embargo, como es normal en cualquier metodología, toma en cuenta los conocimientos extra usados para el desarrollo de la aplicación que no forman parte del modelo de ninguna de las etapas del OOHDM, un ejemplo de esto pueden ser los conocimientos de diseño gráfico. La implementación de esta metodología conlleva diferentes formas de desarrollo en cada una de las etapas mencionadas anteriormente, por lo que se procederá a dar un cuadro general de lo realizado con esta metodología y el desarrollo del proyecto.

El método emplea diferentes campos para describir problemas que ocurren de forma cotidiana y describe la solución a dicho problema, de esta forma la solución encontrada puede ser usada repetidamente en diversos procesos y contextos al desarrollar una aplicación.

Una vez expuestas las posibles soluciones que se obtendrán durante la ejecución del método, se puede exigir el desarrollo de una aplicación con la que el creador gane conocimientos, y se prevea el avance o diseño de una tercera parte, estos conocimientos pueden ir desde lo más básico como el diseño de interfase de usuario hasta aplicaciones hipermedia avanzadas. Se puede decir que dentro de las comunidades de desarrolladores se colecciona de forma constante métodos y modelos de diseño y desarrollo más convenientes para la creación de nuevas aplicaciones y plataformas. Esta metodología busca como principal objetivo el crear un grupo de modelos de inter relacionado conciso, para a través de ellos poder desarrollar diseños completos con la aplicación de los esquemas en el sistema que se desea obtener.

La OOHDM forma estos modelos o esquemas en tres grupos que son:

- Modelo Arquitectónico
- Modelo Navegacional
- Modelo de Interfaz

2.2.1. Modelos arquitectónicos

Estos modelos dan puntos de referencia para la implementación hipermedia de la aplicación o plataforma. Los esquemas desarrollados en esta categoría son similares a los expuestos en [29] donde se busca deslindar los problemas de navegación de otro tipo de conductas o acciones del sistema que no tengan que ver con la arquitectura propia de la plataforma o aplicación. Para lograr correctamente esto los modelos arquitectónicos separan en jerarquías los diferentes tipos de enlaces, nodos y acciones.

2.2.2. Modelos navegacionales

Estos modelos o esquemas ayudan a mantener una correcta organización en la estructura de navegación de la plataforma o aplicación hipermedia, de este modo mantener un curso de acciones coherente para los usuarios de dicha implementación. Se podría considerar que el éxito de la plataforma depende en gran porcentaje de estos modelos, ellos muestran en correcto curso de acciones a tomar para los usuarios, y estos siempre mantendrán consciencia de donde y que están haciendo al usar la aplicación [32].

2.2.3. Modelos de interfaz

Los esquemas o modelos de interfaz se pueden considerar un tema casi separado de los modelos anteriores, son abstractos e independientes del ambiente que se usara para el desarrollo e implementación de la plataforma o aplicación hipermedia.

Aunque no lo aparente el diseño de la interfaz gráfica es una de las etapas más complicadas de esta metodología, además de la funcionalidad de los elementos se toma en cuenta la relación que estos tendrán entre sí y con el entorno en el que se desempeñan; y, las relaciones en cantidad y espacialidad [33].

Una vez establecidos los modelos anteriores se aclara que, en el método OOHDM no se proporciona un identificador claro de cómo debe tratarse el manejo de las cuentas de los usuarios, así como perfiles y tareas que se realicen en relación a estos, pese a esto, estos elementos se toman como parte del diseño navegacional para ser considerados dentro de los parámetros que debe seguir una estructura ordenada y funcional.

Se ha definido por algunos autores [34] el uso de panoramas en donde el usuario es el centro absoluto de la aplicación (*Centered Users*) y de esta forma identificar de mejor manera las clases de usuarios y las funciones que este realiza, pudiendo así permitir que la plataforma o aplicación hipermedia lo apoye en dichas actividades.

El método mencionado se inicia con un esquema o modelo conceptual creado por el desarrollador, solo tomando en cuenta las funciones que este piensa deben ser implementadas acorde con su comprensión del dominio.

Para realizarlos con mayor certeza se plantean escenarios para cada uno de los tipos de usuario, de esta manera podemos extrapolar rutas de navegación parciales, una vez obtenidos todas estas rutas de navegación se procede a identificar los puntos de coincidencia entre los varios esquemas, lo que nos permite elaborar un modelo de navegación y un esquema de clases navegacionales estable.

Conforme como avanza la investigación con el método OOHDM se han ido identificando puntos clave en los cuales se implementa un modelo de seguridad y permitir un acceso controlado a cada una de las funciones de acuerdo con su tipo de usuario. Este modelo de seguridad toma en cuenta a cada uno de los usuarios y el contexto en el que este trabaja [35].

Entre algunos de los problemas más relevantes se puede mencionar la necesidad de construir varias funciones o métodos de acción que permitan al usuario mantener una experiencia fluida dentro de la aplicación, se puede ejemplificar uno de estos problemas con la función creada para el control de las consultas dependientes que permiten la plataforma manejar opciones dependiendo de la acción que realice el usuario.

CAPITULO III. DESARROLLO DEL MODELO E IMPLEMENTACION DE LA PLATAFORMA ON LINE

3.1. PRIMERA ETAPA. OBTENCIÓN DE REQUERIMIENTO

3.1.1. Tópicos a ser atendidos

Los Tópicos o puntos principales que fueron atenidos en la plataforma son:

- Búsqueda de servicios
- Promoción de Servicios
- Contacto de Servicios
- Clasificación de servicios

Cada uno de ellos realizado de forma virtual dentro y desde la plataforma *online* desarrollada.

3.1.2. Tópicos relacionados entre sí

La relación existente entre los tópicos resalta a la vista, la plataforma *online* busca generar un espacio en el cual sus usuarios puedan promocionar sus servicios, así como también los puedan buscar en de forma cómoda en un mismo lugar, previniendo la calidad del servicio gracias a los datos solicitados para la promoción de estos y a la posibilidad de realizar una comparativa entre ellos gracias al sistema de calificación de los mismos, así como la facilidad de ser informado al instante de nuevas solicitudes de servicio o de la atención a solicitudes realizadas.

3.1.3. Usuarios a ser atendidos

Los usuarios se dividirán en tres categorías principales de las cuales dos tendrán principal actuación dentro de la plataforma *online* y una que se encargará de la administración de esta, los usuarios serán:

- Usuarios No ofertantes
- Usuarios Ofertantes
- Usuario(s) Administrador(es)

3.1.4. Tareas principales a ser abordadas

Las tareas principales fueron el registro de los usuarios, la búsqueda eficiente de usuarios ofertantes dentro de la plataforma, notificación instantánea al momento de solicitar un servicio, la geolocalización de los usuarios ofertantes y el sistema de calificación.

El registro del usuario se trata con cautela pues se busca mantener los datos del usuario de forma segura para lo cual se trabajará con un token de encriptamiento para mantener todos los datos del usuario seguros.

La búsqueda de servicios se dividirá en dos puntos los usuarios cercanos a tu posición determinados por el área postal registrada y el sistema de geolocalización y un punto de búsqueda general en donde encontraras los usuarios ofertantes registrados en todo el país. La notificación se generará vía correo electrónico y mensajes de texto al momento de realizarse la solicitud en la plataforma.

La geolocalización se encargará de proporcionar la localización de los usuarios en caso de querer movilizarse hasta el punto en el que el ofertante sitúa su oficina o centro de atención.

El sistema de calificación proporcionara noción de la calidad del servicio, así como la factibilidad de trabajar con el usuario no ofertante solicitante.

3.1.5. Tareas y categorías de usuarios

El registro de usuarios correspondió a los dos tipos de usuarios tanto ofertantes como no ofertantes tendrán que realizar el registro para poder acceder a los servicios de la plataforma *online* sin embargo ambos tendrán características diferentes que se cumplirán al registrase.

La búsqueda de usuarios ofertantes está estrechamente ligada a los usuarios no ofertantes quienes serán capaces de navegar de forma libre por las diferentes opciones de búsqueda; no obstante, los usuarios ofertantes también podrán realizar la búsqueda de servicios con la diferencia que se estableció un limitante de acceso a su cuenta en los servicios de búsqueda.

3.1.6. Diagramas de casos de uso

La descripción e identificación de los casos de uso de una aplicación permitió definir los comportamientos del sistema de forma más clara, y obtener los requerimientos funcionales del sistema desarrollo. Estos diagramas definen el grupo de propósitos para los que los actores pueden usar el sistema. El diagrama de casos de uso también es acompañado de sus respectivas plantillas donde se documentó cada

uno añadiendo la información extra como son las precondiciones, flujo normal de actividad, flujo alternativo, postcondiciones, comentarios entre otros.

Actores

Los actores son entidades extrañas al sistema que pueden interactuar con él. Los usuarios pueden ser de diversos tipos como humanos, hardware externo o inclusive otros sistemas. Los usuarios tienen nombres dependiendo el usuario puede comunicarse con el sistema a través de estímulos o eventos. En el caso de la plataforma se cuenta con tres actores que son:

Ilustración 1, Usuario atendidos



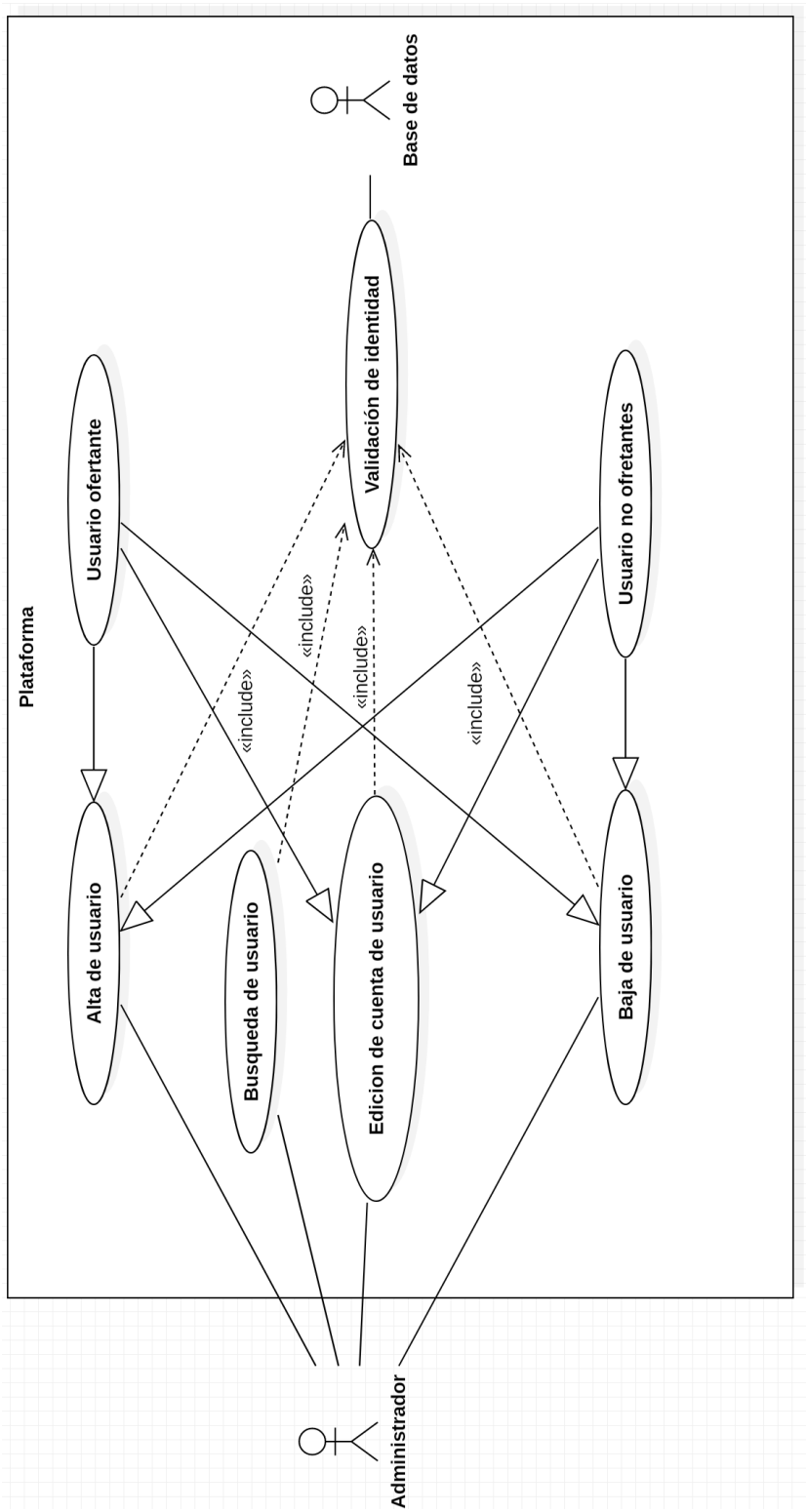
Fuente: Elaboración Propia

Los casos de uso son las secuencias completas de acciones de los actores, en las que se detallan la interacciones entre los actores y el sistema; estos pueden ser sencillos o simples involucrando una sola interacción con un actor o pueden ser complejos incluyendo muchas interacciones con uno muchos actores

3.1.6.1. Gestión de administradores

Diagrama de caso de uso – usuario administrador

Ilustración 2, UCD - Usuarios administrador



Fuente: Elaboración Propia

Tabla 1, Baja de Usuario Administrador,

UC001 Baja de Usuario por Usuario Administrador		
Caso de uso	Baja de usuario por usuario administrador	
Actor(es)	Usuario administrador	
Descripción	Eliminar al usuario seleccionado para impedir su acceso al sistema	
Precondición	Usuario seleccionado debe estar previamente registrado en la plataforma	
	Usuario Seleccionado excede un porcentaje de quejas y denuncias	
	Usuario seleccionado incumple con las condiciones del sitio	
	Acción Actor	**Acción Sistema**
Flujo Principal	1. El Actor selecciona la opción gestión de usuarios.	2. Abre un formulario con los datos de todos los usuarios
	3. El Actor selecciona el tipo de usuario general	4. Despliega un listado con todos los usuarios.
	5. El Actor selecciona eliminar el usuario	6. Pide confirmación de la acción
	7. El Actor confirma la acción	8. Se borra el registro del usuario
Flujo Alternativo	7. El Actor cancela la acción	8. El sistema sale del listado y no se realiza ninguna acción
Flujo Alternativo 2	7. El Actor confirma la acción	8. ocurre un error y la acción no se realiza informando del error al usuario

Fuente: Elaboración Propia

Tabla 2, Alta de usuario por Usuario Administrador

UC002 Alta de Usuario por Usuario Administrador		
Caso de uso	Alta de usuario por usuario administrador	
Actor(es)	Usuario administrador	
Descripción	Crea un nuevo usuario para permitir el acceso a la plataforma	
Precondición	El nuevo usuario no debe estar previamente registrado en la plataforma	
	Acción Actor	**Acción Sistema**
Flujo Principal	1. El Actor selecciona la opción gestión de usuarios.	2. Abre un formulario con los datos de todos los usuarios
	3. El Actor selecciona el tipo de usuario general	4. Despliega un listado con todos los usuarios.
	5. El Actor selecciona crear el usuario	6. Pide confirmación de la acción
	7. El Actor confirma la acción	8. Despliega formularios de ingreso de datos
	9. El Actor completa la información solicitada por el sistema	
	10. El actor confirma la creación	11.Crea en nuevo usuario
Flujo Alternativo	7. El Actor cancela la acción	8. El sistema sale del listado y no se realiza ninguna acción
Flujo Alternativo 2	7. El Actor confirma la acción	8. ocurre un error y la acción no se realiza informando del error al usuario

Fuente: Elaboración Propia

Tabla 3, Edición de Cuanta por Usuario Administrador

UC003 Edición de Cuenta por Usuario Administrador		
Caso de uso	Edición de cuenta por usuario administrador	
Actor(es)	Usuario Administrador	
Descripción	El Actor modifica los datos asociados a un usuario del sistema.	
Precondición	El Actor debe haberse registrado previamente en el sistema.	
	El Actor debe haber proporcionado un nombre de usuario y contraseña válidos	
	El Actor debe tener los permisos necesarios para realizar modificaciones en la información de los usuarios.	
	Acción Actor	**Acción Sistema**
Flujo Principal	1. El Actor selecciona la sección de gestión de usuarios.	2. Se mueve al formulario con los datos de todos los usuarios
	3. El Actor selecciona la opción de modificar del usuario a seleccionado.	4. Despliega Formulario de datos del usuario
	5. El Actor modifica los datos del usuario.	
	6. El Actor confirma la acción	7. el sistema verifica que la información este correcta y completa
		8. Almacena la información.
Flujo Alternativo	6. El Actor cancela la acción	7. El sistema sale del formulario y no se realiza ninguna acción
Flujo Alternativo 2	6. El Actor confirma la acción	7. La información esta incorrecta o incompleta y se muestra un mensaje

Fuente: Elaboración Propia

Tabla 4, Búsqueda de Usuario No Administrador

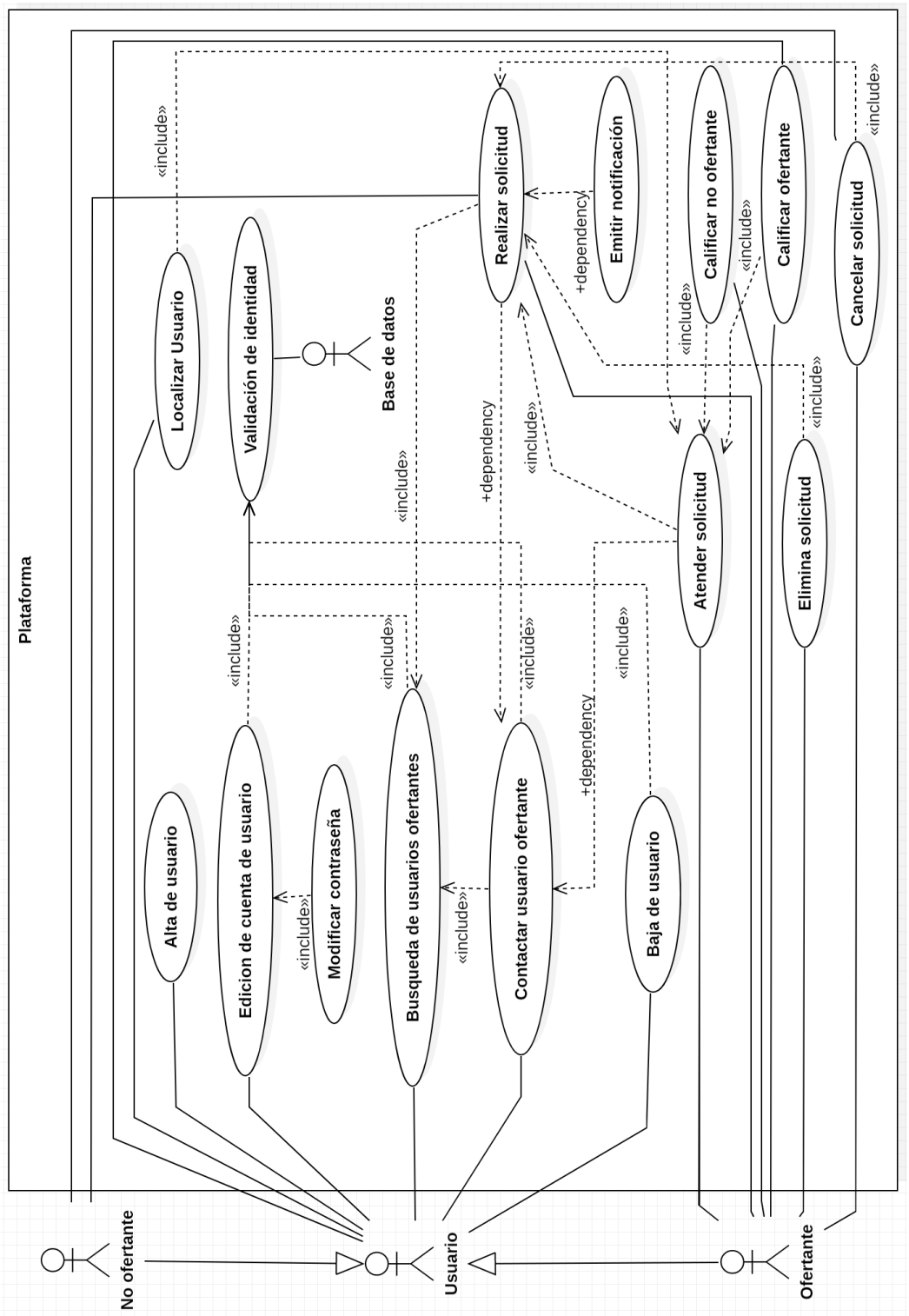
UC004 Búsqueda de Usuario Ofertantes / No ofertantes		
Caso de uso	Búsqueda de usuario ofertantes / no ofertantes	
Actor(es)	Usuario administrador	
Descripción	El Actor registrado ingresa a la plataforma y selecciona la opción buscar de la barra de menú y selecciona las características del usuario que desea encontrar.	
Precondición	El Actor debe estar registrado en la plataforma.	
	El Actor debe iniciar sección exitosa en la plataforma.	
	Acción Actor	**Acción Sistema**
Flujo Principal	1. El Actor selecciona la opción buscar en la barra de menú.	2. Abre el formulario de búsqueda de usuarios ofertantes.
	3. El Actor selecciona la opción de búsqueda que concuerda con sus necesidades	4. Muestra el formulario con todos los usuarios ofertantes que coincidan con las especificaciones ingresadas.
	5. El Actor selecciona uno de los usuarios ofertantes.	9. Despliega el formulario de datos del usuario ofertante.
Flujo Alternativo	7. El Actor cancela la acción	8. El sistema sale del formulario y no se realiza ninguna acción

Fuente: Elaboración Propia

3.1.6.2. Gestión de Cuentas de usuarios ofertantes y no ofertantes

Diagrama de caso de uso – usuarios ofertantes y no ofertantes

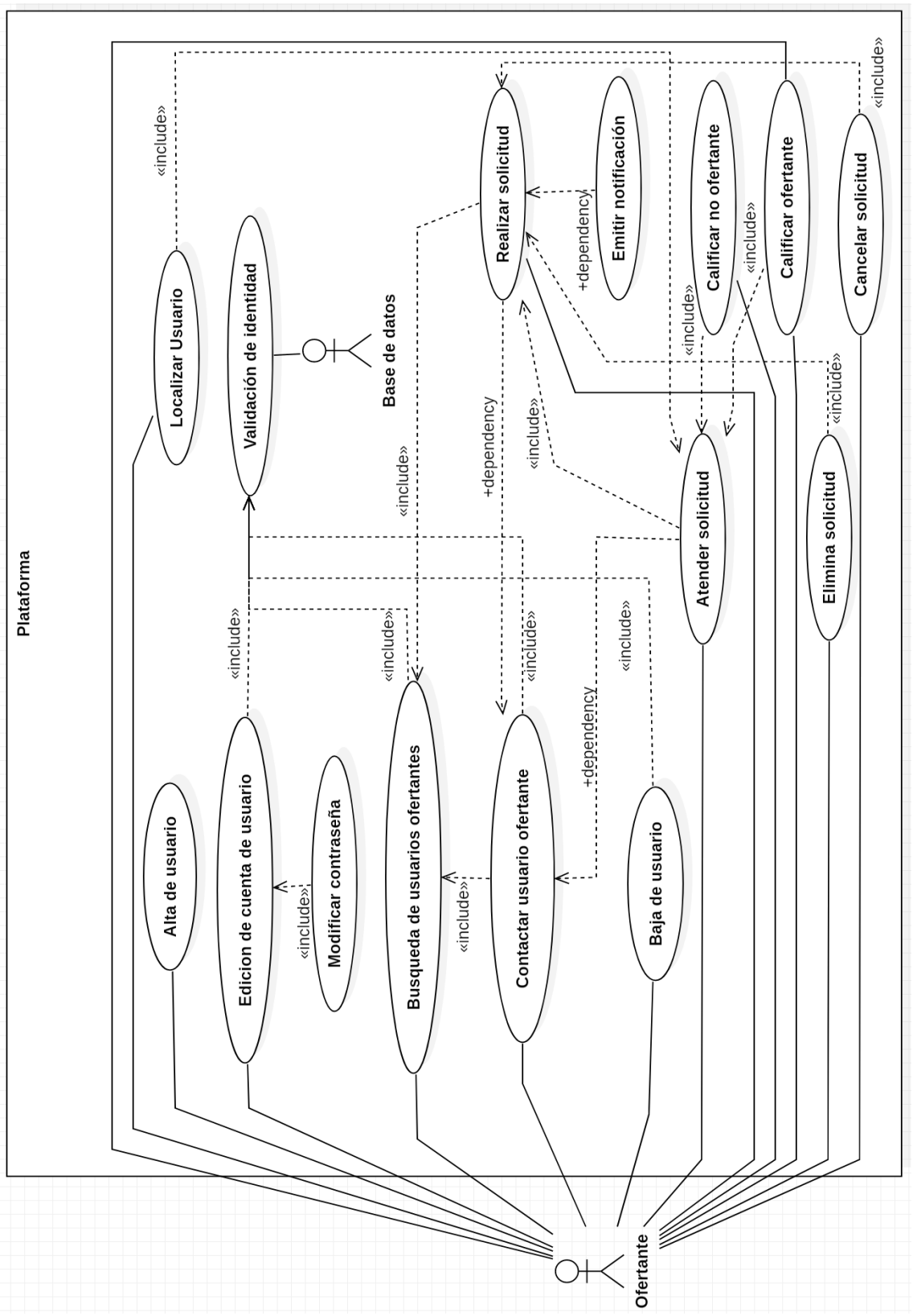
Ilustración 3, UCD - Usuarios



Fuente: Elaboración Propia

Diagrama de caso de uso – usuarios ofertantes

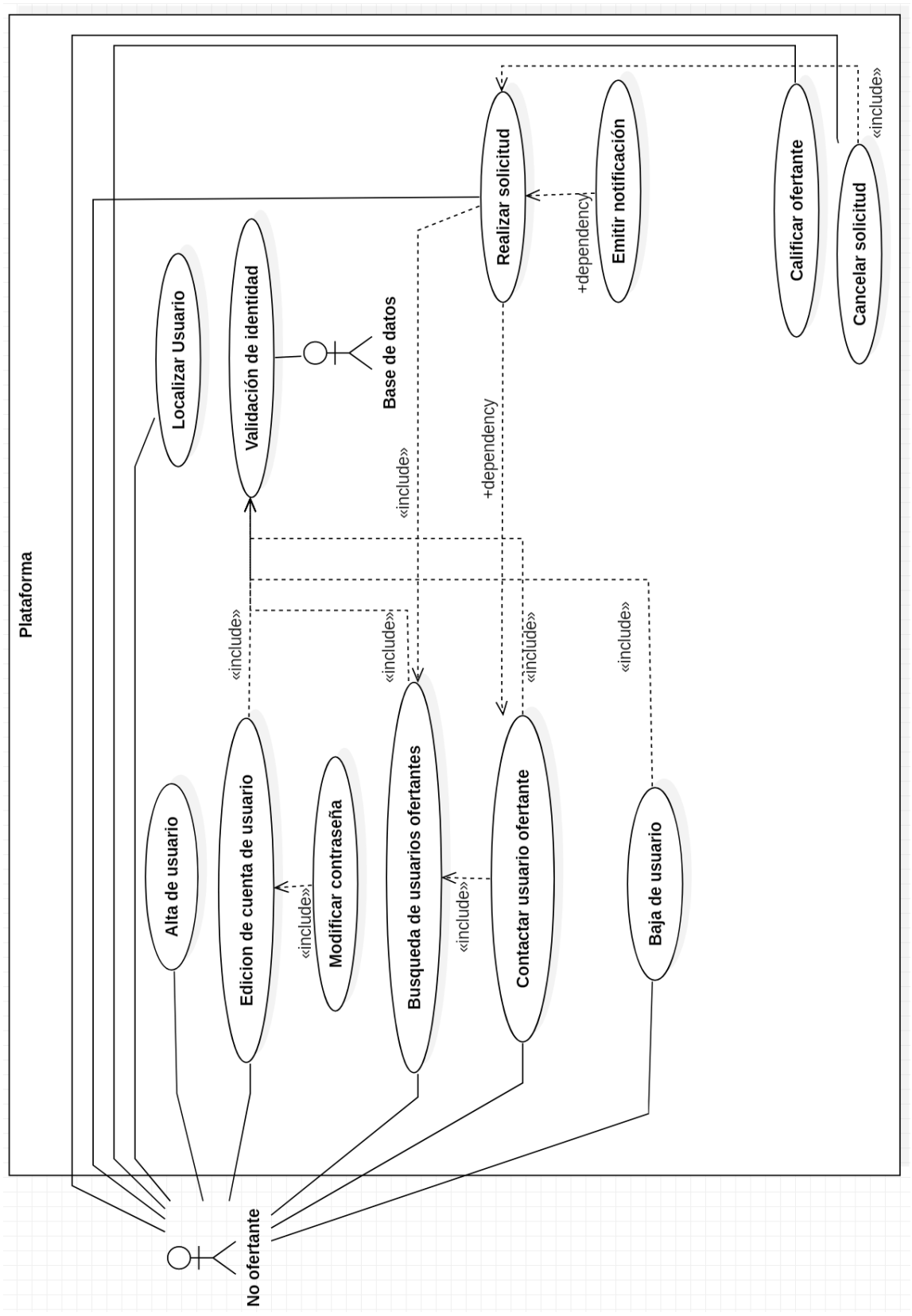
Ilustración 4, UCD - Usuarios ofertantes



Fuente: Elaboración Propia

Diagrama de caso de uso – usuarios no ofertantes

Ilustración 5, UCD – Usuarios no ofertantes



Fuente: Elaboración Propia

Tabla 5, Alta de Usuario ofertantes

UC005 Alta de Usuario ofertantes		
Caso de uso	Alta de usuarios ofertantes	
Actor(es)	Usuarios ofertantes	
Descripción	El actor ingresa los datos requeridos por la plataforma para considerarse registrado en ella.	
Precondición	El actor no tiene que estar registrado en la plataforma	
	El actor del sistema no esta registrado como administrador de la plataforma.	
	Acción Actor	**Acción Sistema**
Flujo Principal	1. El Actor selecciona la opción de registrase la plataforma.	2. Abre el módulo de opciones de registro.
	3. El Actor selecciona la opción de registro de ofertantes	4. Despliega Formulario de registro.
	5. El Actor ingresa sus datos en el formulario de registro.	
	6. El Actor confirma la acción presionando el botón de guardado.	7. Envía un email con un link de confirmación al correo usado para registrarse.
	8. El Actor confirma el registro en la plataforma dándole clic al link recibido.	9. Direcciona al Actor al formulario de inscripción de datos final.
	10. El Actor completa sus datos en el formulario de inscripción de datos final.	
	11. El Actor confirma la acción presionando el botón de Finalizar.	12. Almacena la información y registra el nuevo actor.
		13. Re direcciona al Actor a la página de bienvenida.
Flujo Alternativo	6. El Actor cancela la acción	7. El sistema sale del formulario y no se realiza ninguna acción
Flujo Alternativo 2	6. El Actor confirma la acción	7. La información esta incorrecta o incompleta y se muestra un mensaje

Fuente: Elaboración Propia

Tabla 6, Baja de Usuario ofertante

UC006 Baja de Usuario ofertante		
Caso de uso	Baja de Usuario ofertante	
Actor(es)	Usuario ofertante	
Descripción	El Actor ingresa a su cuenta de usuario y selecciona la opción cerrar cuenta ingresa su contraseña y la cuenta es eliminada.	
Precondición	El Actor esta previamente registrado en la plataforma	
	Acción Actor	**Acción Sistema**
Flujo Principal	1. El Actor ingresa a su cuenta en la plataforma ingresando su nombre de usuario y contraseña.	2.Verifica los datos e ingresa
	3. El Actor selecciona la opción cerrar cuenta.	4. Solicita ingreso de contraseña para confirmar la acción.
	5. El Actor Digita la contraseña y acepta.	6. Verifica su contraseña y elimina la cuenta.
Flujo Alternativo	5. El Actor cancela la acción	6. El sistema sale del formulario y no se realiza ninguna acción
Flujo Alternativo 2	5. El Actor confirma la acción	6. La información está incorrecta o incompleta y se muestra un mensaje

Fuente: Elaboración Propia

Tabla 7, Alta de Usuario no ofertantes

UC007 Alta de Usuario no Ofertantes		
Caso de uso	Alta de usuario no ofertante	
Actor(es)	Usuarios no ofertantes	
Descripción	El actor ingresa los datos requeridos por la plataforma para considerarse registrado en ella.	
Precondición	El actor no tiene que estar registrado en la plataforma	
	El actor del sistema no está registrado como administrador de la plataforma.	
	Acción Actor	**Acción Sistema**
Flujo Principal	1. El Actor selecciona la opción de registrase en la barra de menú de la plataforma.	2. Abre el módulo de opciones de registro.
	3. El Actor selecciona la opción de registro de no ofertantes	4. Despliega Formulario de registro.
	5. El Actor ingresa sus datos en el formulario de registro.	
	6. El Actor confirma la acción presionando el botón de guardado.	7. Envía un email con un link de confirmación al correo usado para registrarse.
	8. El Actor confirma el registro en la plataforma dándole clic al link recibido.	9. Direcciona al Actor al formulario de inscripción de datos final.
	10. El Actor completa sus datos en el formulario de inscripción de datos final.	
	11. El Actor confirma la acción presionando el botón de Finalizar.	8. Almacena la información y registra el nuevo actor.
		13. Redirecciona al Actor a la página de bienvenida.
Flujo Alternativo	6. El Actor cancela la acción	7. El sistema sale del formulario y no se realiza ninguna acción
Flujo Alternativo 2	6. El Actor confirma la acción	7. La información esta incorrecta o incompleta y se muestra un mensaje

Fuente: Elaboración Propia

Tabla 8, Baja de Usuario no ofertante

UC008 Baja de Usuario no Ofertante		
Caso de uso	Baja de usuario no ofertante	
Actor(es)	Usuario no ofertante	
Descripción	El Actor ingresa a su cuenta de usuario y selecciona la opción cerrar cuenta ingresa su contraseña y la cuenta es eliminada.	
Precondición	El Actor esta previamente registrado en la plataforma	
	Acción Actor	**Acción Sistema**
Flujo Principal	1. El Actor se ingresa a su cuenta en la plataforma ingresando su nombre de usuario y contraseña.	2.Verifica los datos e ingresa
	3. El Actor selecciona la opción cerrar cuenta.	4. Solicita ingreso de contraseña para confirmar la acción.
	5. El Actor Digita la contraseña y acepta.	6. Verifica su contraseña y elimina la cuenta.
Flujo Alternativo	5. El Actor cancela la acción	6. El sistema sale del formulario y no se realiza ninguna acción
Flujo Alternativo 2	5. El Actor confirma la acción	6. La información esta incorrecta o incompleta y se muestra un mensaje

Fuente: Elaboración Propia

Tabla 9, Reporte de usuario

UC09 Contactar Usuario Ofertante		
Caso de uso	Contactar usuario ofertante	
Actor(es)	Usuarios no ofertantes / Usuario ofertante	
Descripción	El actor selecciona la opción de contactar un usuario Ofertante en el buscador de personal de la plataforma.	
Precondición	El actor está registrado en la plataforma	
	El actor inicia sesión en la plataforma	
	Acción Actor	**Acción Sistema**
Flujo Principal	1. El Actor selecciona la opción de buscar en la plataforma.	2. Se dirige el módulo de opciones de Búsqueda.
	3. El Actor selecciona la opción de búsqueda de su preferencia.	4. Despliega Formulario de personal disponible en la plataforma.
	5. El Actor ingresa selecciona el usuario en el formulario de personal disponible en la plataforma.	6. Despliega el formulario de perfil del usuario seleccionado.
	8. El Actor confirma la acción presionando el botón de contactar.	9. Envía un mensaje de notificación al usuario seleccionado.
		10. Enlista al usuario seleccionado en la sección de Pendientes.
Flujo Alternativo	8. El Actor cancela la acción	9. El sistema sale del formulario y no se realiza ninguna acción
Flujo Alternativo 2	8. El Actor confirma la acción	9. La notificación no es atendida y no se puede contactar.

Fuente: Elaboración Propia

Tabla 10, Búsqueda de usuario ofertante

UC010 Búsqueda de Usuario Ofertante		
Caso de uso	Búsqueda de usuario ofertante	
Actor(es)	Usuario no ofertante /Usuario Ofertante	
Descripción	El Actor registrado ingresa a la plataforma y selecciona la opción buscar de la barra de menú y selecciona las características del usuario que desea encontrar.	
Precondición	El Actor está registrado en la plataforma.	
	El Actor inicia sección exitosa en la plataforma.	
	Acción Actor	**Acción Sistema**
Flujo Principal	1. El Actor selecciona la opción buscar en la barra de menú.	2. Abre el formulario de búsqueda de usuarios ofertantes.
	3. El Actor selecciona la opción de búsqueda que concuerda con sus necesidades	4. Muestra el formulario con todos los usuarios ofertantes que coincidan con las especificaciones ingresadas.
	5. El Actor selecciona uno de los usuarios ofertantes.	6. Despliega el formulario de datos del usuario ofertante.
Flujo Alternativo	5. El Actor cancela la acción	6. El sistema sale del formulario y no se realiza ninguna acción

Fuente: Elaboración Propia

Tabla 11, Solicitud de servicio

UC0011 Realizar Solicitud de Servicio		
Caso de uso	Realizar solicitud de servicio	
Actor(es)	Usuario no ofertante /Usuario Ofertante	
Descripción	El actor da clic en el botón de solicitar servicio del usuario ofertante	
Precondición	El actor está registrado en la plataforma	
	El actor realiza una búsqueda previa	
	El actor selecciona el servicio que desea solicitar	
	Acción Actor	**Acción Sistema**
Flujo Principal	1. El Actor selecciona la opción contactar	2. Genera la solicitud y la enlista en la sección me solicitan de usuario solicitado.
		3. Enlista la solicitud realizada en la sección solicitudes del actor
		4. Envía un correo de notificación al usuario solicitado
		5. Envía un mensaje de texto de notificación al usuario solicitado
Flujo Alternativo	6. El Actor no realiza ninguna acción	

Fuente: Elaboración Propia

Tabla 12, Cancelar solicitud de servicio

UC0012 Cancelar Solicitud de Servicio		
Caso de uso	Cancelar solicitud de servicio	
Actor(es)	Usuario no ofertante /Usuario Ofertante	
Descripción	El actor cancela una solicitud de servicio realizada	
Precondición	El actor está registrado en la plataforma	
	El actor tiene solicitudes pendientes de respuesta	
	Acción Actor	**Acción Sistema**
Flujo Principal	1. El Actor da clic en el botón pendientes.	2. Recorre las secciones hasta llegar a la sección pendientes
	3. El Actor selecciona la opción solicite de la sección pendientes	4. Se muestra el formulario con las solicitudes pendientes de respuesta
	5. El Actor selecciona el botón cancelar correspondiente en el formulario	6. Se elimina la solicitud de la sección pendiente de los usuarios involucrados
Flujo Alternativo	6. El Actor no realiza ninguna acción	

Fuente: Elaboración Propia

Tabla 13, Calificar usuario ofertante

UC0013 Calificar Usuario Ofertante		
Caso de uso	Calificar usuario ofertante	
Actor(es)	Usuario no ofertante /Usuario Ofertante	
Descripción	El Actor califica el servicio obtenido en una escala del 1 al 5 donde 1 es pésimo y 5 es excelente	
Precondición	El Actor está registrado en la plataforma	
	El Actor tiene solicitudes de servicio aceptadas	
	El Actor tiene solicitudes pendientes de calificación	
	Acción Actor	**Acción Sistema**
Flujo Principal	1. El Actor da clic en el botón pendientes.	2. Recorre las secciones hasta llegar a la sección pendientes
	3. El Actor selecciona la opción solicite de la sección pendientes	4. Se muestra el formulario con las solicitudes pendientes de calificación
	5. El Actor selecciona la calificación que desea dar al usuario dentro del formulario	6. Se elimina la solicitud de la sección pendiente de los usuarios involucrados
Flujo Alternativo	5. El Actor no proporciona calificación	

Fuente: Elaboración Propia

Tabla 14, Calificar usuario no ofertante

UC0014 Calificar Usuario no Ofertante		
Caso de uso	Calificar usuario no ofertante	
Actor(es)	Usuario Ofertante	
Descripción	El Actor califica al cliente en una escala del 1 al 5 donde 1 es pésimo y 5 es excelente	
Precondición	El Actor esta registrado en la plataforma	
	El Actor acepta las solicitudes de servicio	
	El Actor tiene solicitudes aceptadas pendientes de calificación	
	Acción Actor	**Acción Sistema**
Flujo Principal	1. El Actor da clic en el botón pendientes.	2. Recorre las secciones hasta llegar a la sección pendientes
	3. El Actor selecciona la opción me solicitan de la sección pendientes	4. Se muestra el formulario con las solicitudes pendientes de calificación
	5. El Actor selecciona la calificación que desea dar al usuario dentro del formulario	6. Se elimina la solicitud de la sección pendiente de los usuarios involucrados
Flujo Alternativo	5. El Actor no proporciona calificación	

Fuente: Elaboración Propia

Tabla 15, Atender solicitud de servicio

UC0015 Atender Solicitud de Servicio		
Caso de uso	Atender solicitud de servicio	
Actor(es)	Usuarios ofertantes	
Descripción	El actor da clic en el botón de atender para aceptar la solicitud de servicio del usuario no ofertante	
Precondición	El Actor está registrado en la plataforma	
	El Actor tiene solicitudes de servicio en espera	
	Acción Actor	**Acción Sistema**
Flujo Principal	1. El Actor da clic en el botón pendientes.	2. Recorre las secciones hasta llegar a la sección pendientes
	3. El Actor selecciona la opción me solicitan de la sección pendientes	4. Se muestra el formulario con las solicitudes pendientes de respuesta
	5. El Actor selecciona el botón atender del formulario	6. Envía una notificación al correo electrónico y número de celular de solicitante
Flujo Alternativo	5. El Actor no realiza ninguna acción	

Fuente: Elaboración Propia

Tabla 16, Eliminar solicitud de servicio

UC0016 Eliminar Solicitud De Servicio		
Caso de uso	Eliminar solicitud de servicio	
Actor(es)	Usuarios ofertantes	
Descripción	El actor elimina una solicitud de servicio que le realizaron	
Precondición	El actor está registrado en la plataforma	
	El actor tiene solicitudes pendientes de respuesta	
	Acción Actor	**Acción Sistema**
Flujo Principal	1. El Actor da clic en el botón pendientes.	2. Recorre las secciones hasta llegar a la sección pendientes
	3. El Actor selecciona la opción me solicitan de la sección pendientes	4. Se muestra el formulario con las solicitudes pendientes de respuesta
	5. El Actor selecciona el botón eliminar correspondiente en el formulario	6. Se elimina la solicitud de la sección pendiente de los usuarios involucrados
Flujo Alternativo	5. El Actor no realiza ninguna acción	

Fuente: Elaboración Propia

Tabla 17, Localizar usuario

UC0017 Localizar Usuario		
Caso de uso	Eliminar solicitud de servicio	
Actor(es)	Usuario no ofertante /Usuario ofertante	
Descripción	El actor puede visualizar la localización del usuario de la plataforma	
Precondición	El actor está registrado en la plataforma	
	El actor tiene solicitudes atendidas	
	Acción Actor	**Acción Sistema**
Flujo Principal	1. El Actor da clic en el botón pendientes.	2. Recorre las secciones hasta llegar a la sección pendientes
	3. El Actor selecciona la opción solicite de la sección pendientes	4. Muestra el formulario con las solicitudes.
	5. El Actor selecciona el botón contactar del usuario correspondiente al en el formulario	6. Muestra el formulario con la información y sistema de geolocalización
	7. El Actor selecciona el botón localizar usuario	8. La ubicación de desplaza de la ubicación actual del actor a la ubicación del usuario a localizar
Flujo Alternativo	3. El Actor selecciona la opción me solicitan de la sección pendientes	4. Muestra el formulario con las solicitudes.
	5. El Actor selecciona el botón contactar del usuario correspondiente al en el formulario	6. Muestra el formulario con la información y sistema de geolocalización
	7. El Actor selecciona el botón localizar usuario	8. La ubicación de desplaza de la ubicación actual del actor a la ubicación del usuario a localizar
Flujo Alternativo2	5. El Actor no realiza ninguna acción	

Fuente: Elaboración Propia

Tabla 18, Edición de cuenta de usuario no administrador

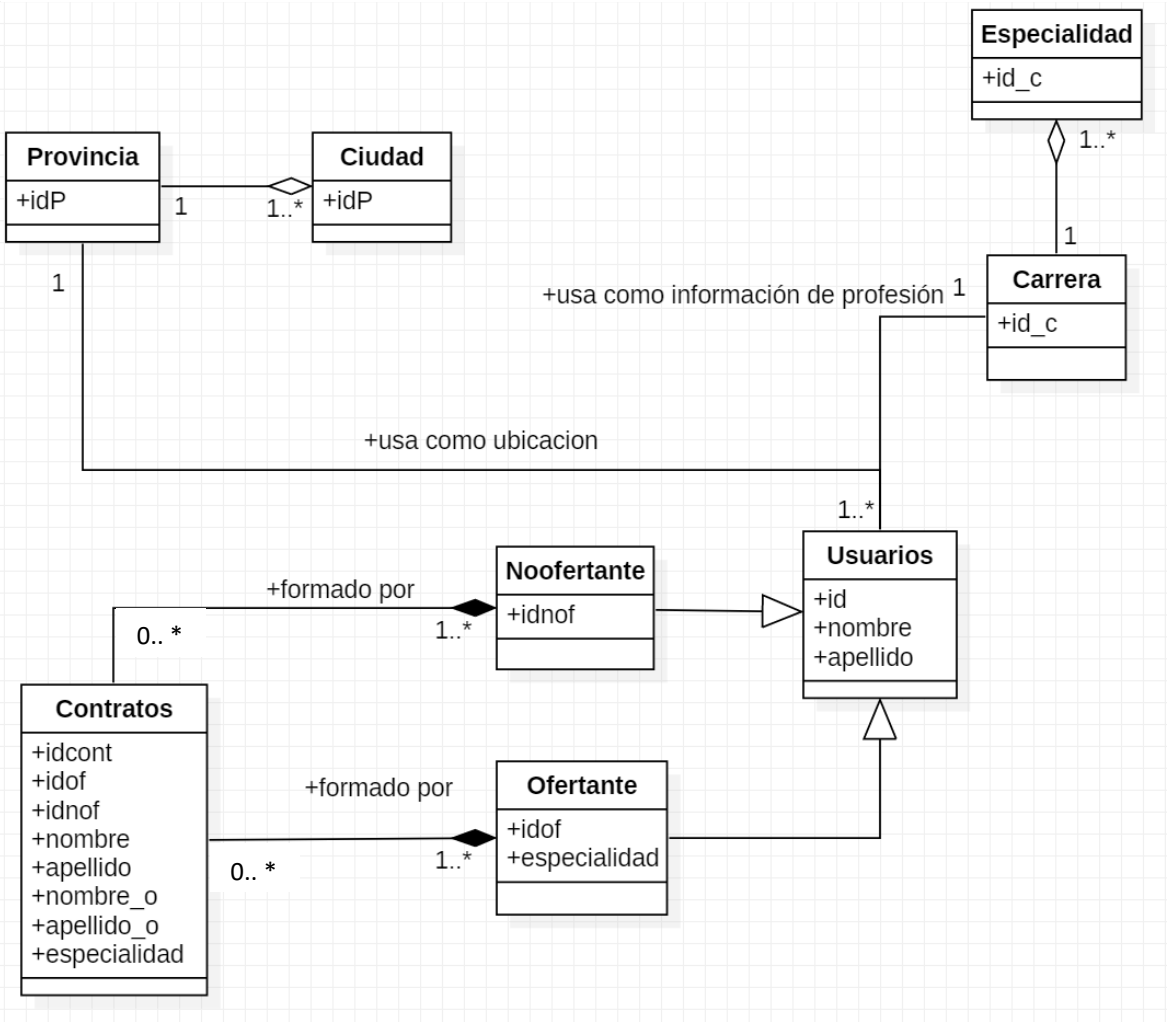
UC018 Edición De Cuenta De Usuario No Administrador		
Caso de uso	Edición de cuenta de usuario no administrador	
Actor(es)	Usuario no ofertante /Usuario ofertante	
Descripción	El Actor modifica los datos asociados a su cuenta en el sistema.	
Precondición	El Actor se registró previamente en el sistema.	
	El Actor proporcionó un nombre de usuario y contraseña válidos	
	El Actor tiene los permisos necesarios para realizar modificaciones en la información de los usuarios.	
0	**Acción Actor**	**Acción Sistema**
Flujo Principal	1. El Actor da clic en el botón pendientes.	2. Recorre las secciones hasta llegar a la sección pendientes
	3. El Actor selecciona la opción solicite de la sección pendientes	4. Se muestra el formulario con las solicitudes pendientes de respuesta
	5. El Actor selecciona la opción cancelar de la solicitud deseada dentro del formulario	6. Se elimina la solicitud de la sección pendiente de los usuarios involucrados
Flujo Alternativo	6. El Actor cancela la acción	7. El sistema sale del formulario y no se realiza ninguna acción

Fuente: Elaboración Propia

3.2. SEGUNDA ETAPA. MODELO CONCEPTUAL

Anteriormente se menciona la función de un modelo o esquema conceptual, a continuación, con efectos de comprender mejor el modelado se presenta la representación gráfica de las clases y relaciones dentro de la plataforma.

Ilustración 6, CRD Modelo conceptual



Fuente: Elaboración Propia

3.2.1. Tbl_carreras

Pertenece a la tabla de carreras disponibles para seleccionar dentro de la plataforma. Dentro de esta tabla se encuentran los campos:

Tabla 19, Tabla Carreras

Nombre del Campo	Tipo	Razón
id_c	Integer	Id de registro
nombre	varchar	Informativo

Fuente: Elaboración Propia

3.2.2. Tbl_especialidad

Pertenece a la tabla de especialidades relacionadas con las sus correspondientes carreras, estas están disponibles para seleccionar dentro de la plataforma. Dentro de esta tabla se encuentran los campos:

Tabla 20, Tabla Especialidades

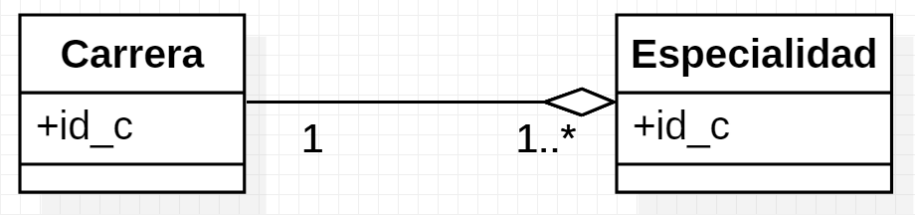
Nombre del Campo	Tipo	Razón
id_es	Integer	Id de registro
id_c	integer	Id de enlace
nombre	varchar	Informativo

Fuente: Elaboración Propia

3.2.3. Diagrama UML Carrera -Especialidad

El diagrama muestra la relación existente entre las dos tablas antes expuestas y su interacción entre sí, donde la clase especialidad pertenecen a la clase carrera y al eliminar la clase especialidad la clase carreara no se ve afectada.

Ilustración 7, CRD Carrera-Especialidad



Fuente: Elaboración Propia

3.2.4. Tbl_provincias

Pertenece a la tabla de provincias disponibles para seleccionar dentro del sistema. Dentro de esta tabla se encuentran los campos:

Tabla 21, Tabla Provincias

Nombre del Campo	Tipo	Razón
idP	Integer	Id de registro
nombre	varchar	Informativo

Fuente: Elaboración Propia

3.2.5. Tbl_ciudades

Pertenece a la tabla de ciudades relacionadas con las correspondientes provincias, estas están disponibles para seleccionar dentro de la plataforma. Dentro de esta tabla se encuentran los campos:

Tabla 22, Tabla Ciudades,

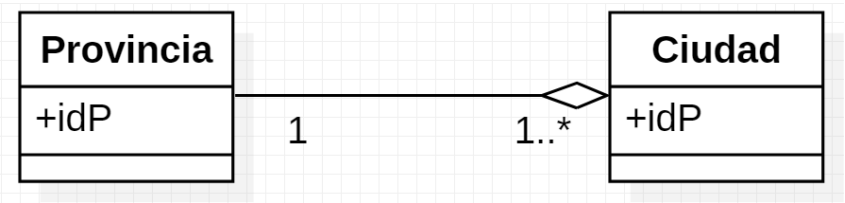
Nombre del Campo	Tipo	Razón
idc	Integer	Id de registro
idP	integer	Id de enlace
nombre	varchar	Informativo

Fuente: Elaboración Propia

3.2.6. Diagrama UML Provincia-Ciudad

El diagrama muestra la relación existente entre las dos tablas antes expuestas y su interacción entre sí, donde la clase ciudad pertenecen a la clase provincia y al eliminar la clase ciudad la clase provincia no se ve afectada.

Ilustración 8, CRD Provincia-Ciudad



Fuente: Elaboración Propia

3.2.7. Tbl_usuarios

Pertenece a la tabla usuarios que se registraran en la plataforma para poder buscar y ofertar servicios en ella. Dentro de esta tabla se encuentran los campos:

Tabla 23, Tabla Usuarios

Nombre del Campo	Tipo	Razón
id	integer	Id de registro
nombre	varchar	Informativo
apellido	varchar	Informativo
correo	varchar	Validación
password	varchar	Validación
last-session	date	Informativo
activacion	integer	Nivel de restricción
token	varchar	Nivel de restricción
token-password	varchar	Nivel de restricción
password-request	varchar	Nivel de restricción
id_tipo	varchar	Nivel de restricción
puntos_c	float	Informativo
puntos_o	float	Informativo
fecha_reg	float	Informativo

Fuente: Elaboración Propia

3.2.8. Tbl_ofertantes

Datos adicionales que pertenecen al tipo ofertantes que promocionan sus servicios dentro de la plataforma y están disponibles para búsqueda y contrato. Dentro de esta tabla se encuentran los campos:

Tabla 24, Tabla Ofertantes

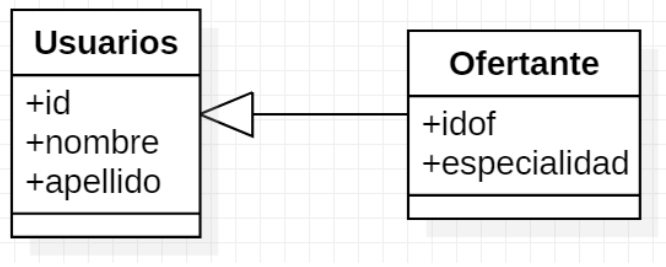
Nombre del Campo	Tipo	Razón
ido	integer	Id de registro
idof	integer	Id de enlace
carrera	varchar	Informativo
especialidad	varchar	Informativo
titulo	varchar	Informativo
telefono1	varchar	Informativo
telefono2	varchar	Informativo
cedula	varchar	Informativo
foto	varchar	Informativo
provincia	varchar	Informativo
ciudad	varchar	Informativo
direccion	varchar	Dato de Georeferencia
postal	integer	Dato de Georeferencia
referencia	varchar	Informativo
lat	doble	Dato de Georeferencia
log	double	Dato de Georeferencia
token	varchar	Nivel de restricción

Fuente: Elaboración Propia

3.2.9. Diagrama UML Usuarios - Ofertantes

El diagrama muestra la relación existente entre las dos tablas antes expuestas y su interacción entre sí, donde la clase ofertantes pertenecen y depende de la clase usuarios y al eliminar la clase usuarios la clase ofertantes también se elimina.

Ilustración 9, CRD Usuarios-Ofertantes



Fuente: Elaboración Propia

3.2.10. Tbl_noofertantes

Datos adicionales que pertenecen al tipo no ofertantes que buscan contratar un servicio de los disponibles para búsqueda y contrato dentro de la plataforma. Dentro de esta tabla se encuentran los campos:

Tabla 25, Tabla No ofertantes,

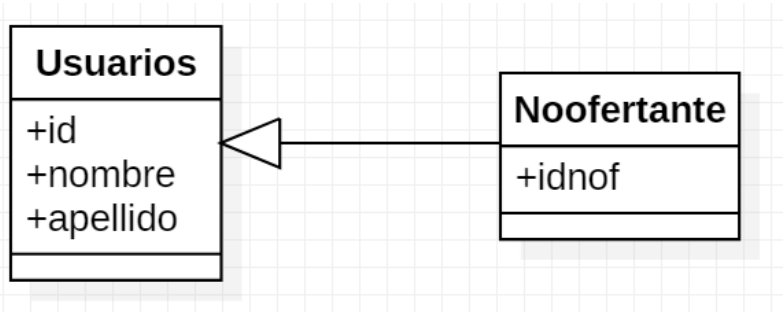
Nombre del Campo	Tipo	Razón
idb	Integer	Id de registro
idnof	Integer	Id de registro
foto	varchar	Informativo
telefono	float	Informativo
provincia	varchar	Informativo
ciudad	varchar	Informativo
direccion	varchar	Dato de Georeferencia
referencia	varchar	Informativo
postal	varchar	Dato de Georeferencia
lat	Integer	Dato de Georeferencia
log	varchar	Dato de Georeferencia
token	varchar	niveles de restricción

Fuente: Elaboración Propia

3.2.11. Diagrama UML Usuarios - Noofertantes

El diagrama muestra la relación existente entre las dos tablas antes expuestas y su interacción entre sí, donde la clase noofertantes es una subclase de la clase usuarios y al eliminar la clase usuarios la clase noofertantes también se elimina.

Ilustración 10, CRD Usuarios-Noofertantes



Fuente: Elaboración Propia

3.2.12. Tbl_contratos

Pertenece a la tabla de Administradores, aquellos usuarios que estarán al tanto del sistema y de las acciones que se deban tomar para el correcto funcionamiento de la plataforma. Dentro de esta tabla se encuentran los campos:

Tabla 26, Tabla Contratos

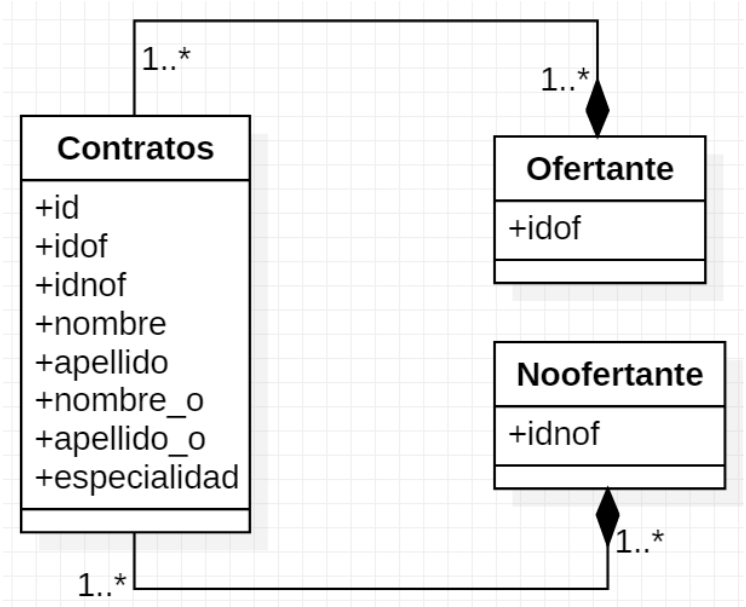
Nombre del Campo	Tipo	Razón
idcon	integer	Id de registro
idnof_c	integer	Id de enlace
nombre	varchar	Informativo
apellido	varchar	Informativo
idof_c	integer	Id de enlace
nombre_o	varchar	Informativo
apellido_o	varchar	Informativo
especialidad	varchar	Informativo
solicitud	integer	niveles derestriccion
confirmacion	integer	niveles derestriccion
fecha_s	datetime	Fecha control
fecha_c	datetime	Fecha control
puntos_c	integer	Informativo
puntos_o	integer	Informativo
comentarios	varchar	Informativo

Fuente: Elaboración Propia

3.2.13. Diagrama UML Contratos - ofertantes y noofertante

El diagrama muestra la relación existente entre las tablas contratos y las tablas de ofertantes y no ofertantes, donde la tabla contratos está compuesta por la información de las tablas mencionadas y su información heredada de la tabla usuarios, fusionándola y combinándola generando el registro de actividades e interacción de los usuarios.

Ilustración 11, CRD Contratos-Usuarios no Administradores



Fuente: Elaboración Propia

3.3. TERCERA ETAPA. ESTRUCTURA NAVEGACIONAL

El diseño de la estructura navegacional en OOHDM pertenecen a la agrupación de modelos o esquemas recopilados durante el desarrollo de la plataforma o aplicación hipermedia.

El diseño navegacional de la plataforma se puede clasificar en:

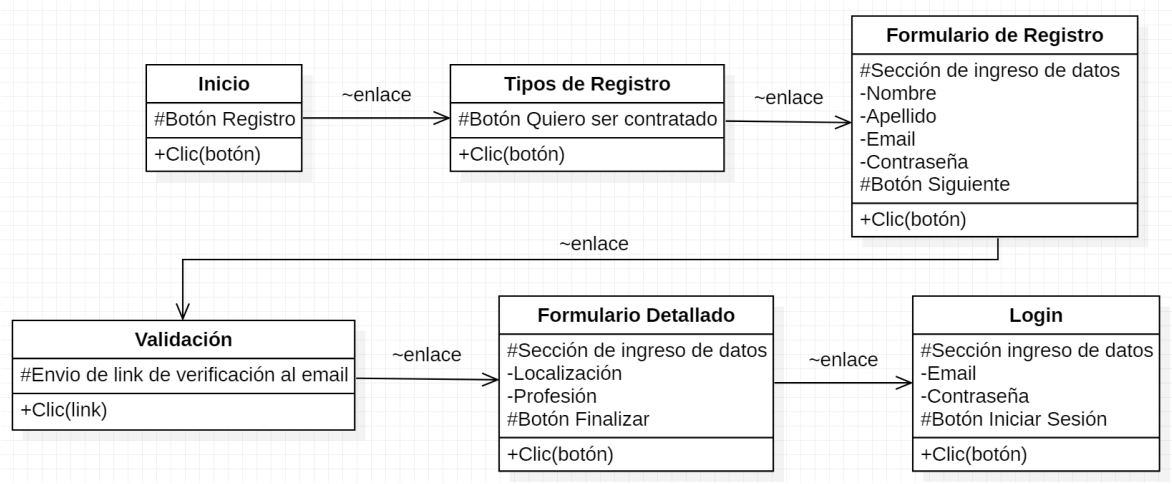
- Nivel de usuarios ofertantes
- Nivel de usuarios no ofertantes
- Nivel de usuarios Administrador
- Diagrama General Navegacional

3.3.1. Niveles usuarios ofertantes

Este nivel comprende el modo de navegación especifico, es decir en donde los usuarios ofertantes poden personalizar sus perfiles o realizar consultas acordes a su nivel permitido.

- Registro Usuario Ofertante

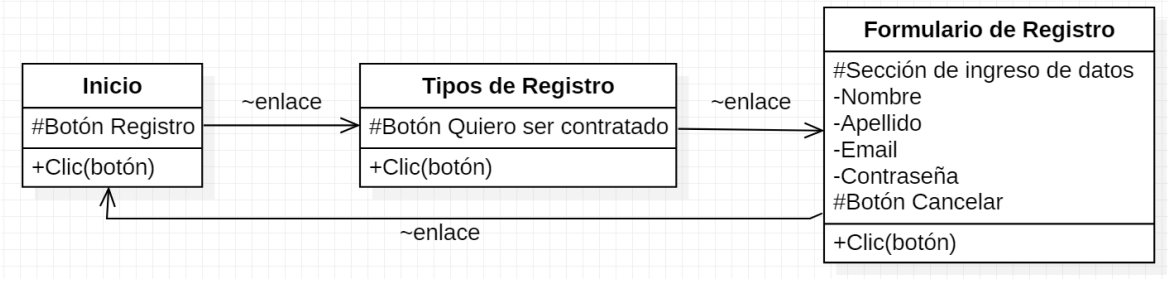
Ilustración 12, NCS Registro Ofertante



Fuente: Elaboración Propia

- Registro Cancelado

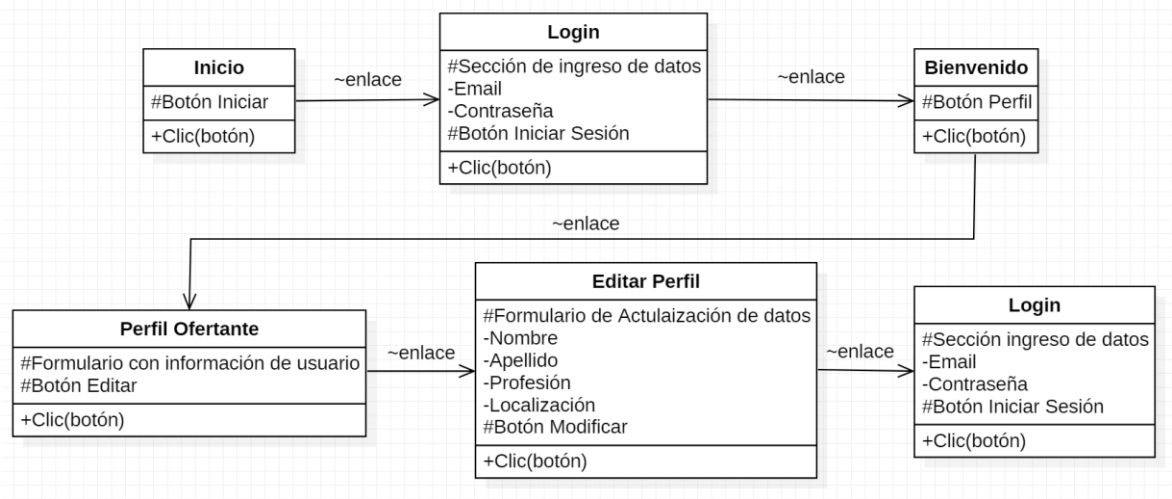
Ilustración 13, NCS Registro cancelado



Fuente: Elaboración Propia

- Editar Datos de Perfil

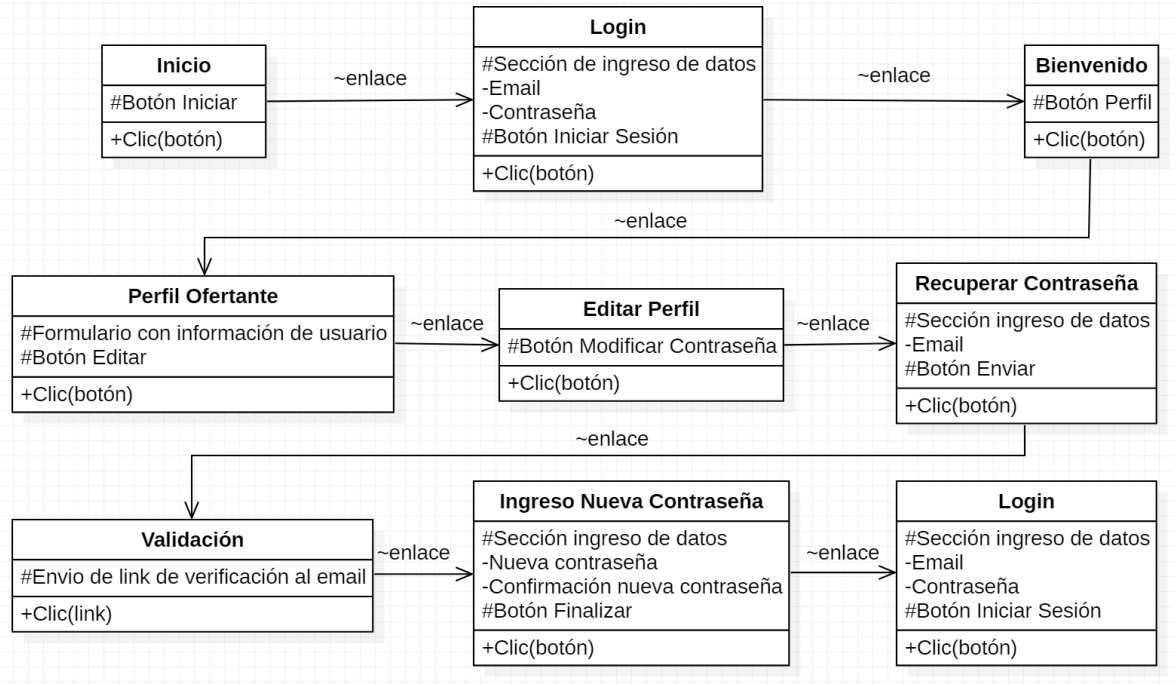
Ilustración 14, NCS Editar perfil



Fuente: Elaboración Propia

- Cambiar Contraseña

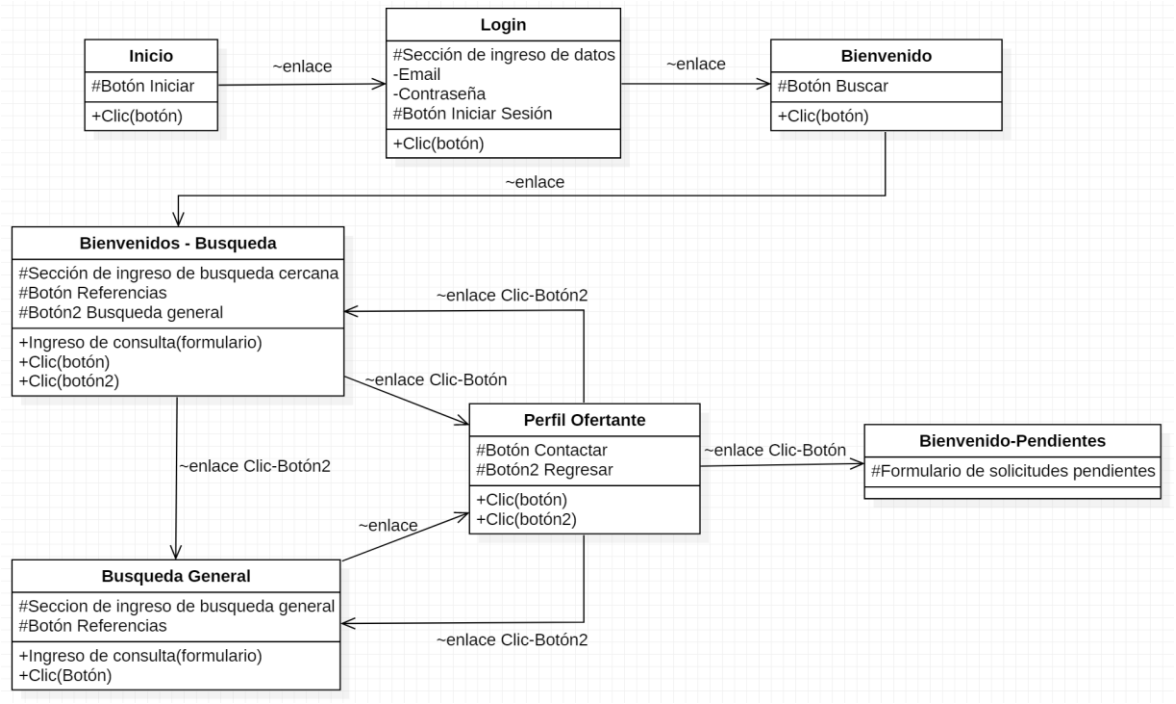
Ilustración 15, NCS Cambiar Contraseña



Fuente: Elaboración Propia

- Búsqueda y Solicitud

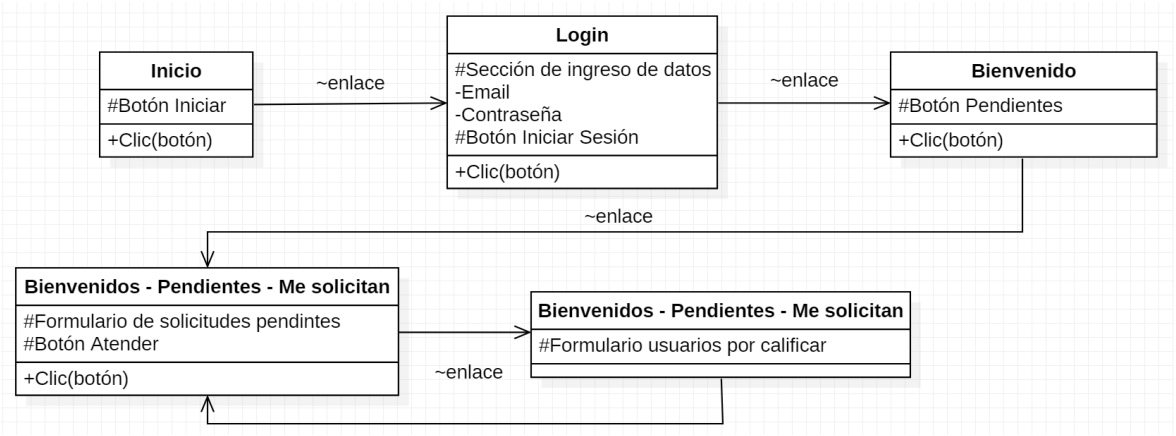
Ilustración 16, NCS Búsqueda y Solicitud



Fuente: Elaboración Propia

- Atender Solicitud de Contacto

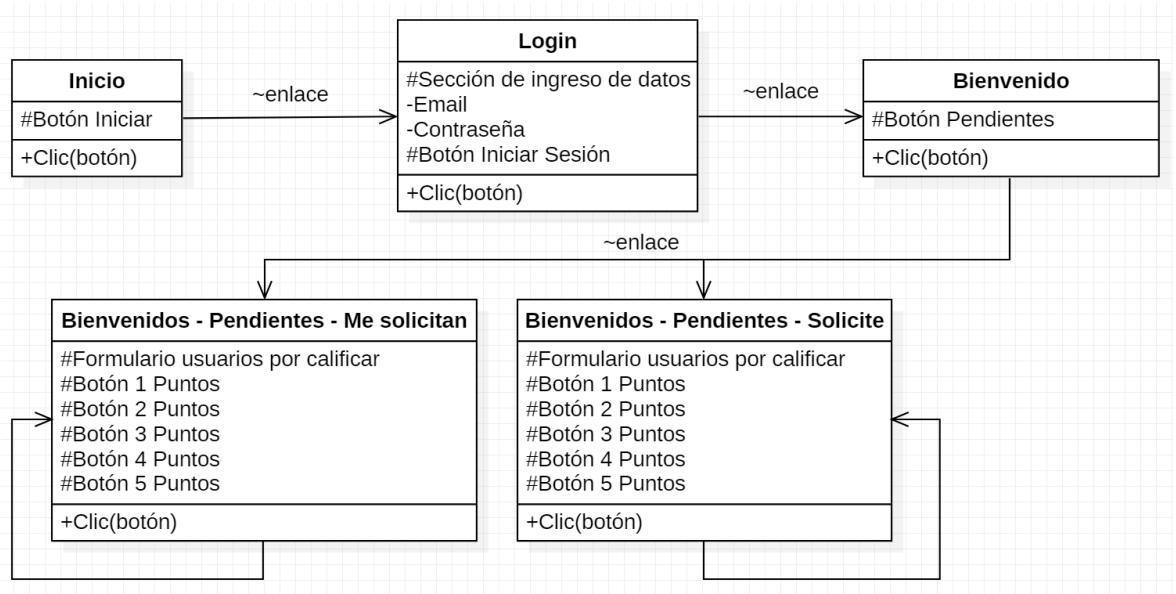
Ilustración 17, NCS Atender Solicitud



Fuente: Elaboración Propia

- Calificar Usuarios Ofertantes y No Ofertantes

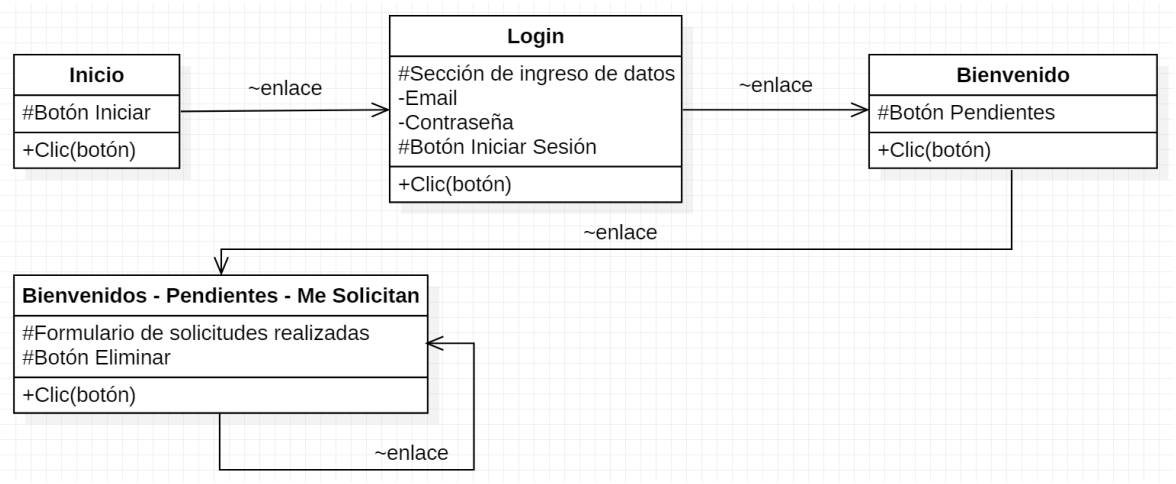
Ilustración 18, NCS Calificar Usuario no Administrador



Fuente: Elaboración Propia

- Eliminar Solicitud de Contacto

Ilustración 19, NCS Eliminar Solicitud



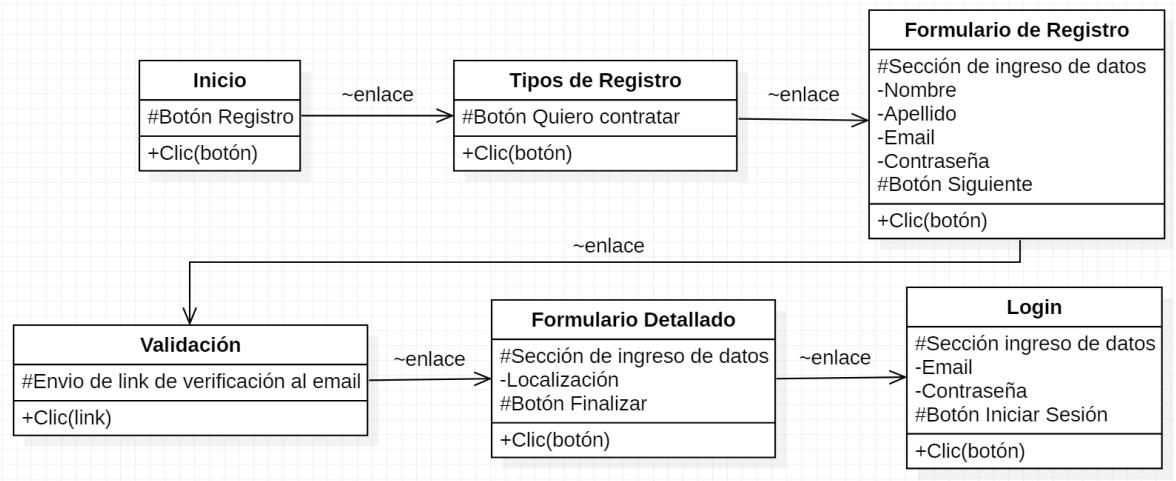
Fuente: Elaboración Propia

3.3.2. Niveles usuarios no ofertantes

Este nivel comprende el modo de navegación básico, es decir en donde los usuarios podrán realizar consultas al sistema.

- Registro No Ofertante

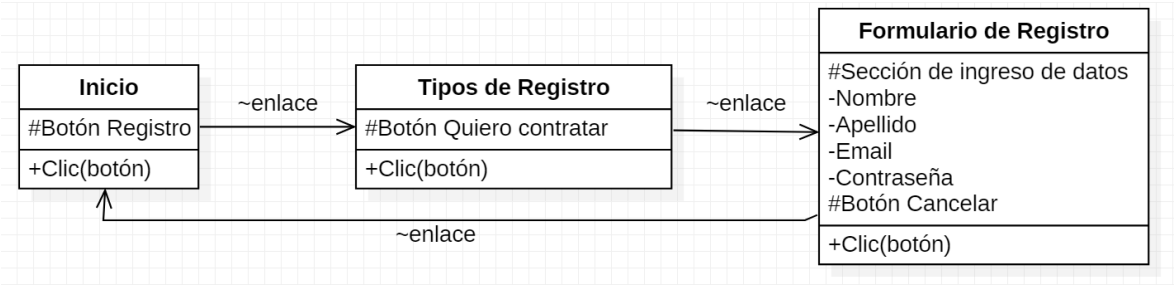
Ilustración 20, NCS Registro no Ofertante



Fuente: Elaboración Propia

- Registro Cancelado

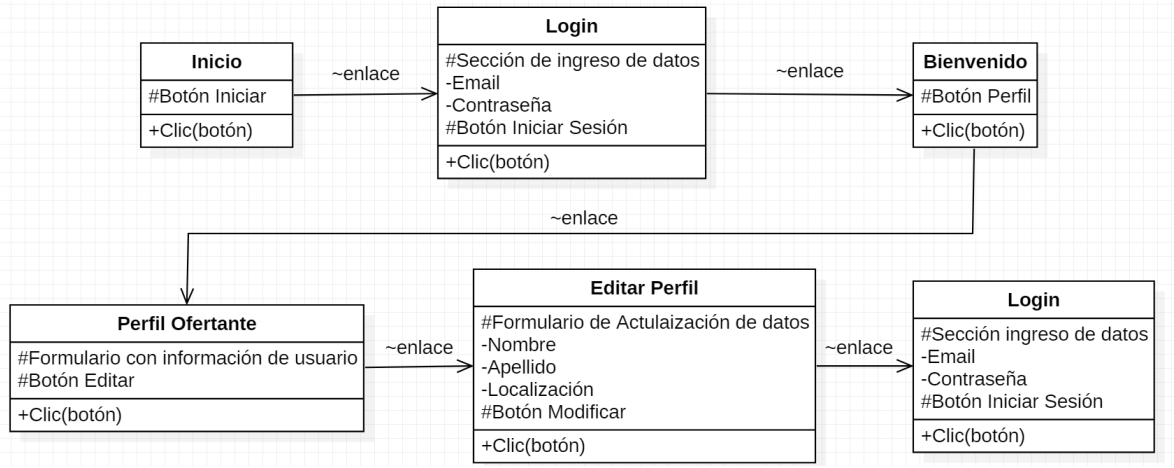
Ilustración 21, NCS Registro Cancelado



Fuente: Elaboración Propia

- Editar Datos de Perfil No Ofertante

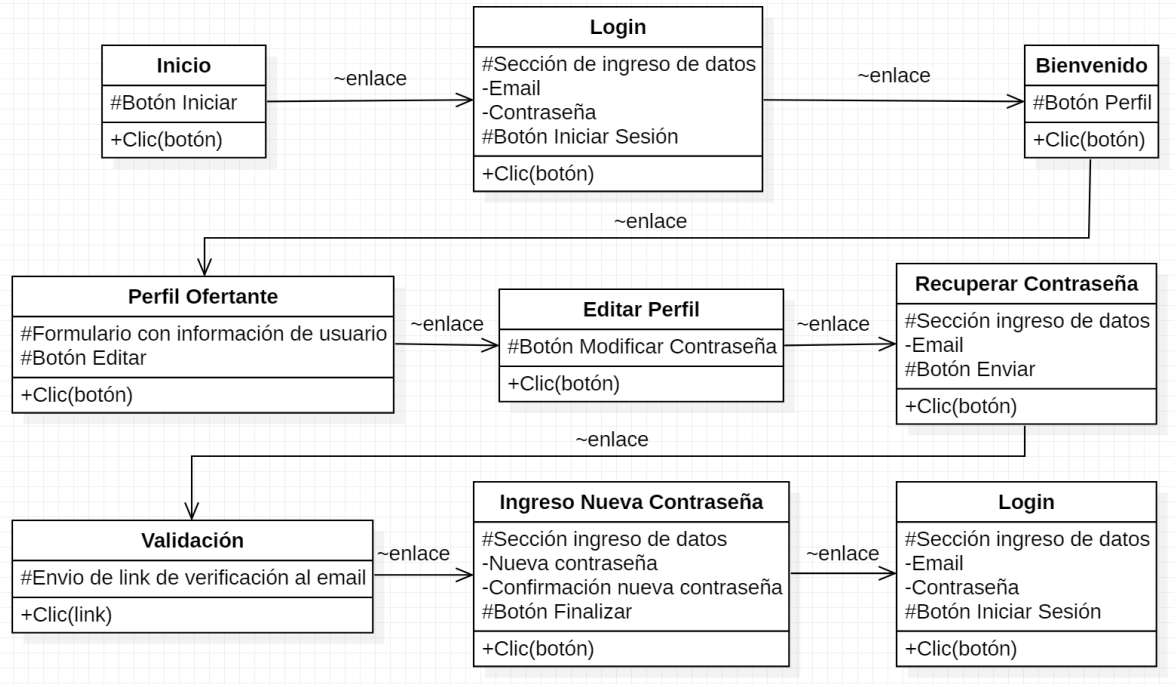
Ilustración 22, NCS Editar Perfil No Ofertante



Fuente: Elaboración Propia

- Cambiar Contraseña

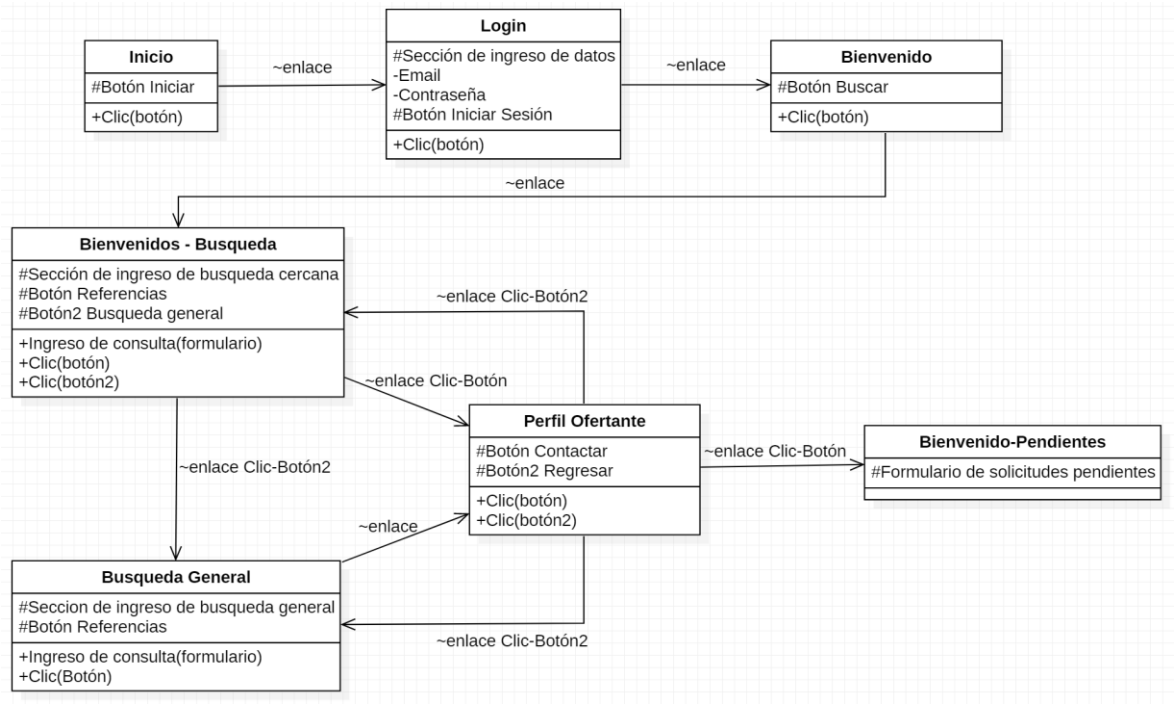
Ilustración 23, NCS Cambiar Contraseña



Fuente: Elaboración Propia

- Búsqueda y Solicitud

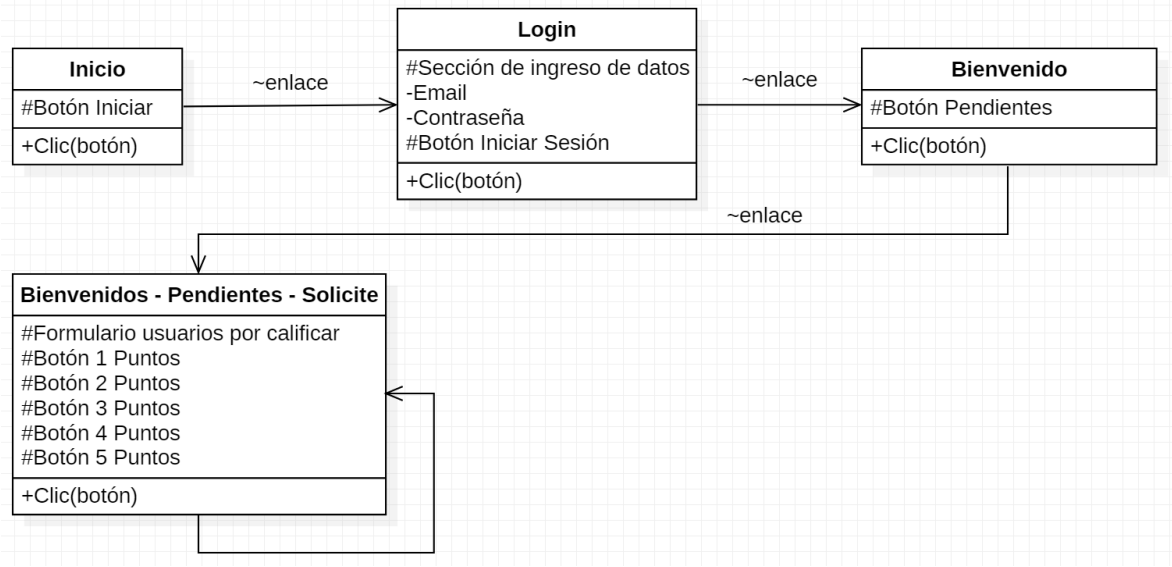
Ilustración 24, NCS Búsqueda y Solicitud



Fuente: Elaboración Propia

- Calificar Ofertante

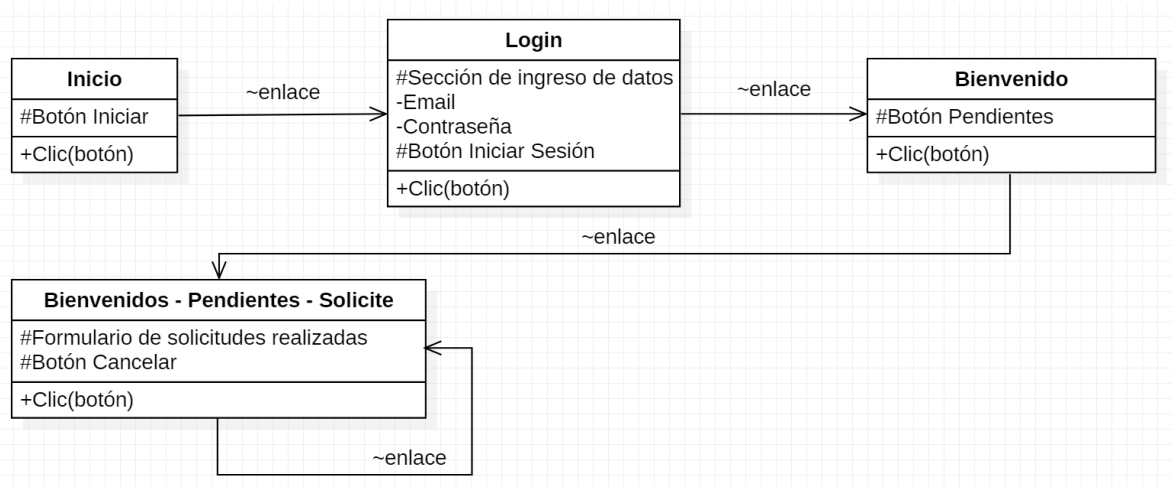
Ilustración 25, NCS Calificar Ofertante



Fuente: Elaboración Propia

- Cancelar solicitud de contacto

Ilustración 26, Cancelar Solicitud de Contacto



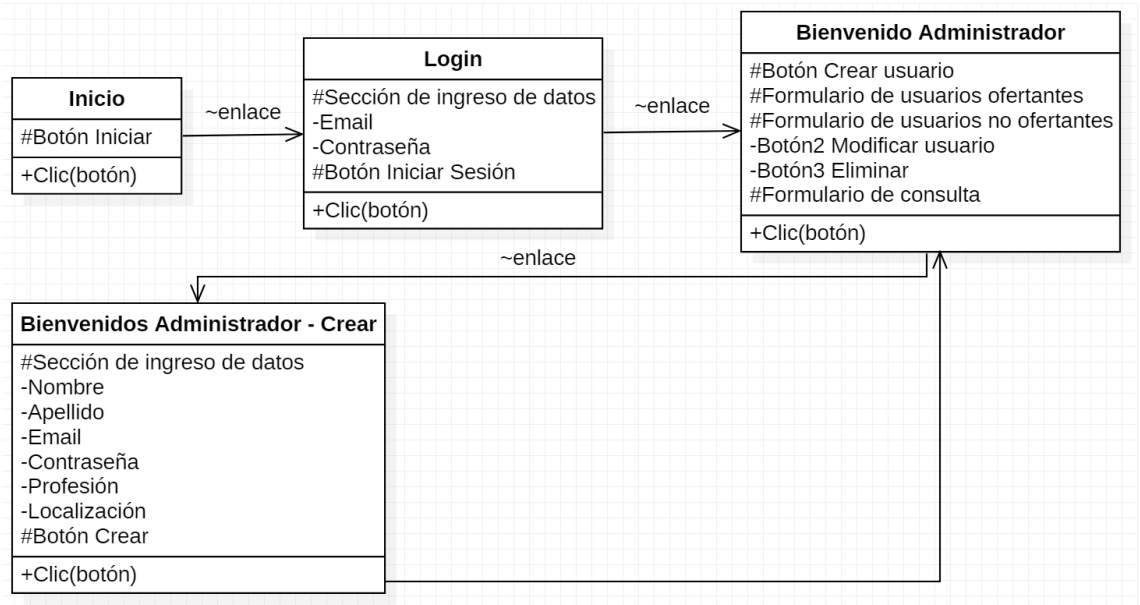
Fuente: Elaboración Propia

3.3.3. Nivel de Administración.

Este nivel es el punto de control del sistema, es decir, donde el administrador podrá realizar las modificaciones al sistema de acuerdo a las funciones del sistema.

- Crear usuario

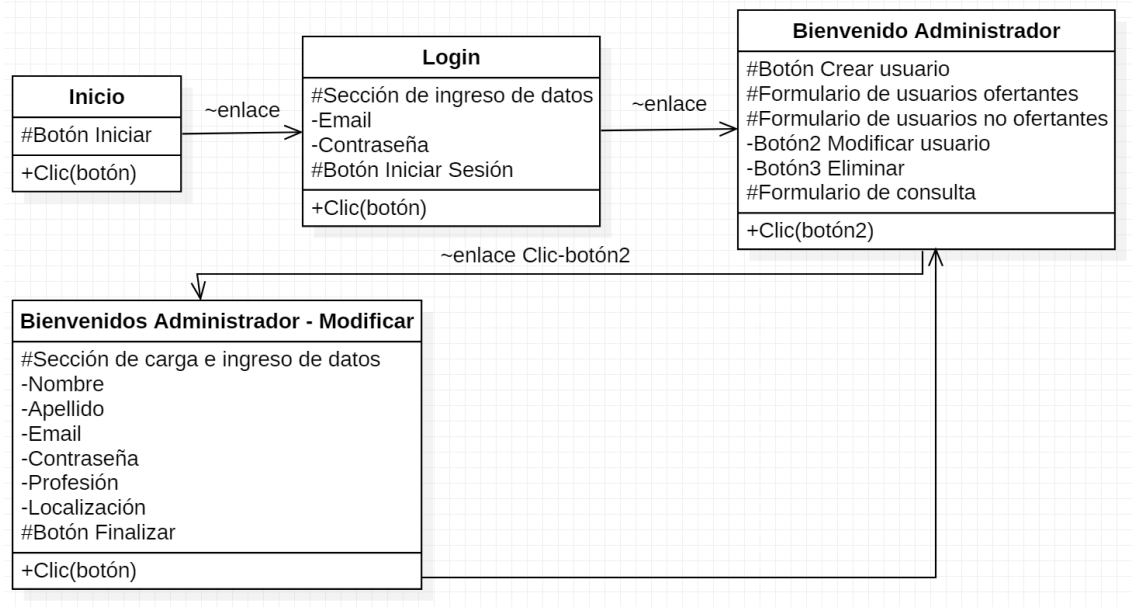
Ilustración 27, NCS Crear Usuario



Fuente: Elaboración Propia

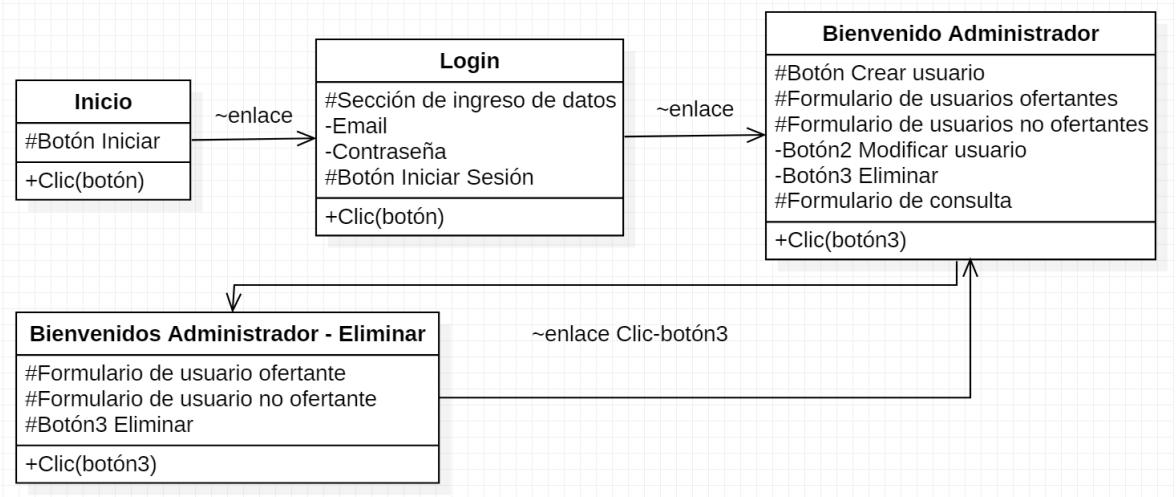
- Modificar usuario

Ilustración 28, NCS Modificar Usuario



- Eliminar Usuario

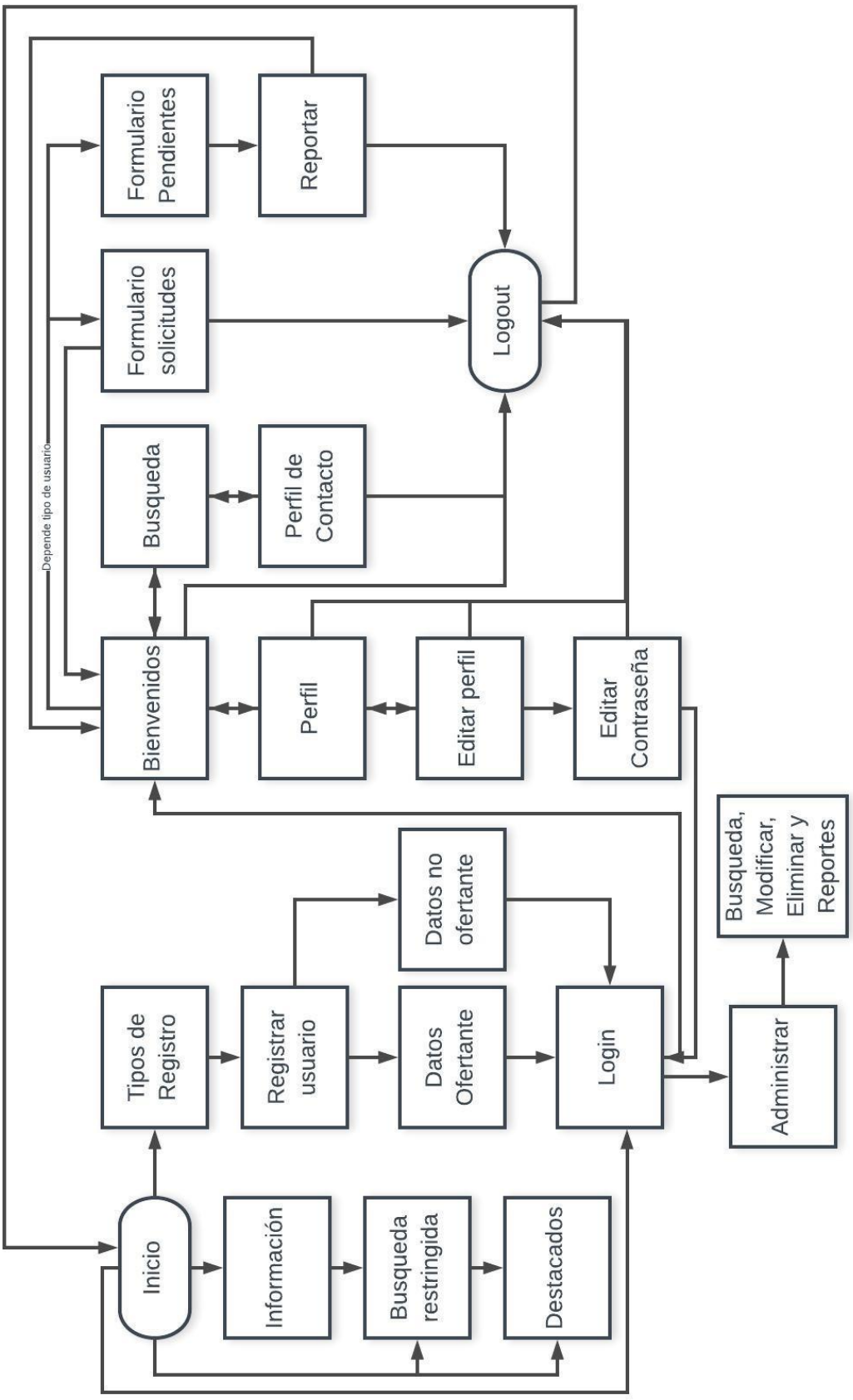
Ilustración 29, NCS Eliminar Usuario



Fuente: Elaboración Propia

3.3.4. Diagrama General Navegacional.

Ilustración 30, Diagrama General Navegacional



Fuente: Elaboración Propia

3.4. CUARTA ETAPA. INTERFAZ ABSTRACTA

3.4.1. Diseño adaptativo o *Responsive Design*

¿Qué es?

El *Responsive Design* diseño adaptativo, es el método de desarrollo que se viene usando desde tiempo atrás para proporcionar una *web* que se adaptada a las diferentes plataformas que se utilizan hoy en día, por ejemplo: ordenador, *tablet* y *Smartphone*.

Su gran mayoría consiste en el uso del lenguaje CSS, que cuenta con un atributo llamado *mediaquery* que permite convertir una página *web* normal en versión una multiplataforma capaz de adaptarse a diversas dimensiones de pantallas existentes, ofreciendo a los usuarios una experiencia más agradable y funcional que cubre las necesidades de estos.

¿Porque usar desarrollo responsivo y no desarrollo específico para dispositivos?

El desarrollo se centra en tres puntos principales del porque se optó por el desarrollo *responsive* en lugar del desarrollo específico para dispositivos.

El primer punto, la no incurrencia en el uso de repositorios de descarga como *Apple Store* y *Play Store*; permite evaluar el posicionamiento inicial de la plataforma manteniendo costes mínimos, lo que lleva al segundo punto.

El segundo punto, se reducen los costes de desarrollo y mantenimiento de la aplicación *web*. La posibilidad de reusar la misma plantilla para varias plataformas reduce la inversión tanto en mantenimiento como en desarrollo, es decir que un cambio en la aplicación *web* repercute en todas las plataformas.

El tercer punto, rapidez en el desarrollo, actualmente existen varios *framework* que permiten el desarrollo e implementación de sitios *web* responsivos de forma rápida y eficaz, entre los *framework* más populares encontramos *Bootstrap*, *Foundation* y *Materialize*.

3.4.2. *Bootstrap* vs. *Foundation* vs. *Materialize*

A continuación, se muestra una tabla comparativa con los aspectos principales considerados más importantes por los usuarios de herramientas de desarrollo como son *Hacker news, Reddit, Stack Overflow Stats, GitHub* y *Stackshare* [36].

Tabla 27, Comparativa de Framework

Framework	Reputación			Codigo Disponible	Flexibilidad	Personalizable
	Hacker news	Reddit	Stack Overflow Stats	GitHub	stackshare	stackshare
Bootstrap	3.29k	6.33k	93.1k	128k	665	362
Fundation	-	1.39k	698	27.7k	-	71
Materialize	-	375	2.61k	34.2k	25	-

Fuente: Elaboración Propia

3.4.3. *Framework Bootstrap*

En base a la comparativa del tema anterior se decidió el usar el *framework Bootstrap* el cual facilito el desarrollo de la plataforma *online*. Bootstrap cuenta con varias características que ayudaron a definir las posiciones y la actuación del entorno de acuerdo a las acciones de los usuarios [37].

Una de las características principales de *bootstrap* permite subdividir la pantalla de desarrollo en 12 secciones las cuales permiten a través de comandos ejecutados en lenguaje HTML5 y CSS3 organizar las pantallas o módulos acorde a las necesidades del usuario, lo que implica un mejor manejo del espacio y redistribución de los elementos según el dispositivo del cual se acceda; cabe recalcar que si bien ayudan con la distribución antes mencionada esta no se considera una distribución *responsive*, para alcanzar dicha cualidad se implementan varios elementos tanto en lenguaje de codificación como en diseño. Una vez finalizado el diseño navegacional, es necesario especificar las diferentes

interfaces de la aplicación. Esto significa definir las diferentes vistas que tendrán antes y después de registrarse.

Para realizar los diferentes módulos de la plataforma se utilizarán varias librerías disponibles en el contenido del framework las cuales al ser usadas en conjunto nos permitirán generar un entorno adecuado para la navegación de los usuarios de la plataforma.

Las librerías que se utilizarán son las librerías de CSS como:

- *bootstrap-grid.css*
- *bootstrap-grid.css.map*
- *bootstrap-grid.min.css*
- *bootstrap-grid.min.css.map*
- *bootstrap-reboot.css*
- *bootstrap-reboot.css.map*
- *bootstrap-reboot.min.css*
- *bootstrap-reboot.min.css.map*
- *bootstrap.css*
- *bootstrap.css.map*
- *bootstrap.min.css*
- *bootstrap.min.css.map*

Que facilitan la creación y diseño del entorno permitiendo lograr con mayor comodidad el paso de la etapa 4 diseño de interfaz abstracta a la etapa 5 de implementación, puesto que, reduciremos tiempo en el detallado de varios elementos de plataforma como en textos sus dimensiones y características o botones con sus respectivas características.

También se usaron las librerías de js como:

- *bootstrap.bundle.js*
- *bootstrap.bundle.js.map*
- *bootstrap.bundle.min.js*
- *bootstrap.bundle.min.js.map*
- *bootstrap.js*
- *bootstrap.js.map*
- *bootstrap.min.js*
- *bootstrap.min.js.map*

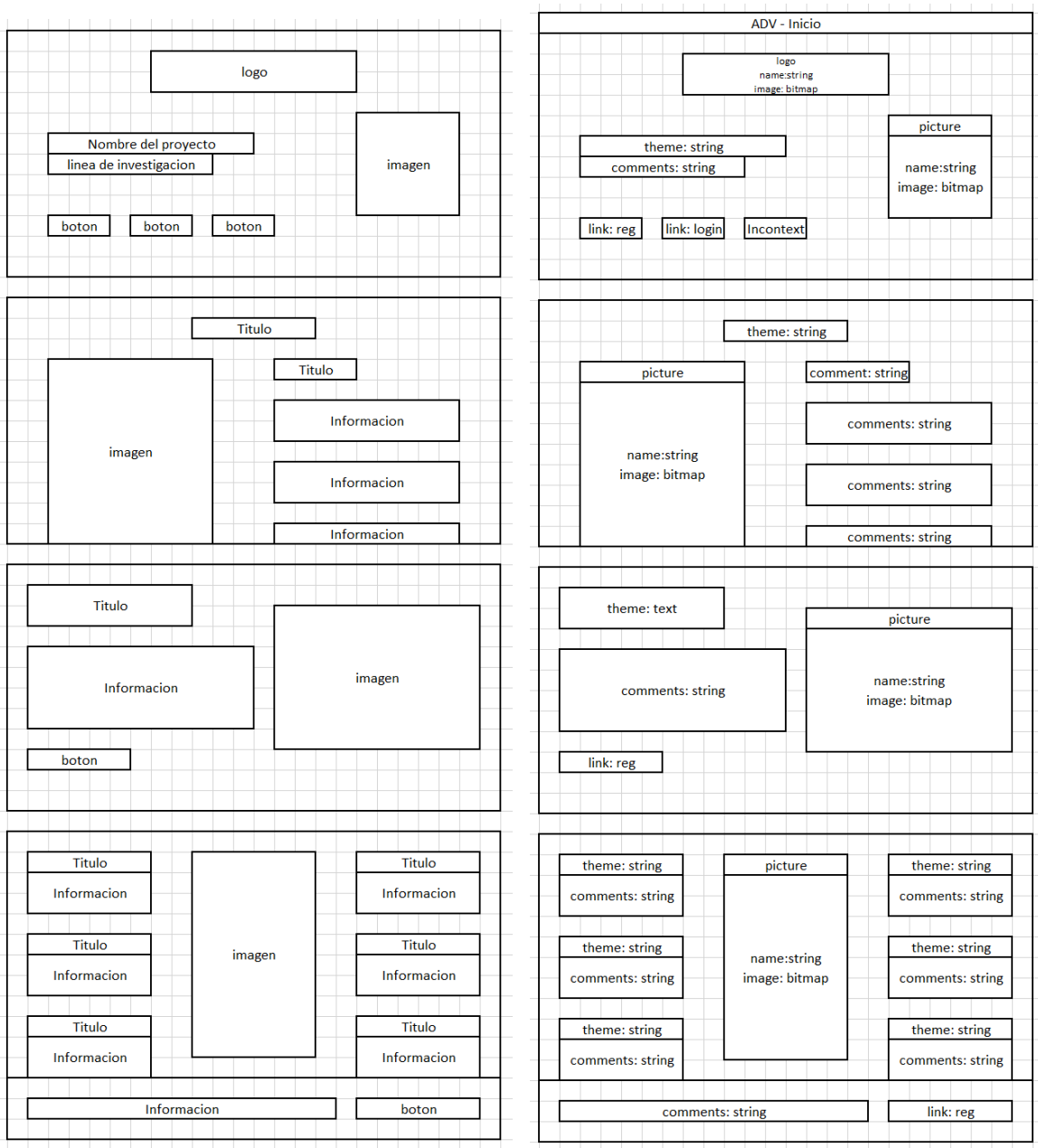
Con estas librerías se logrará dotar de características dinámicas y atractivas para los usuarios de la plataforma permitiendo la creación de una barra de búsqueda dinámica sin necesidad de recargar constantemente la página de búsqueda.

3.4.4. Modulo: Inicio

A continuación, se muestra una representa de la plataforma para los posibles usuarios, en este podrán encontrar los accesos para registrarse en la plataforma y para iniciar sesión en ella si ya estas registrado, también se muestra información sobre para que fue desarrollada y como funciona.

Registrar una cuenta en la plataforma lleva al usuario a explorar los siguientes módulos que definen que tipo de usuario está integrando la plataforma.

Ilustración 31, ADV Modulo Inicio

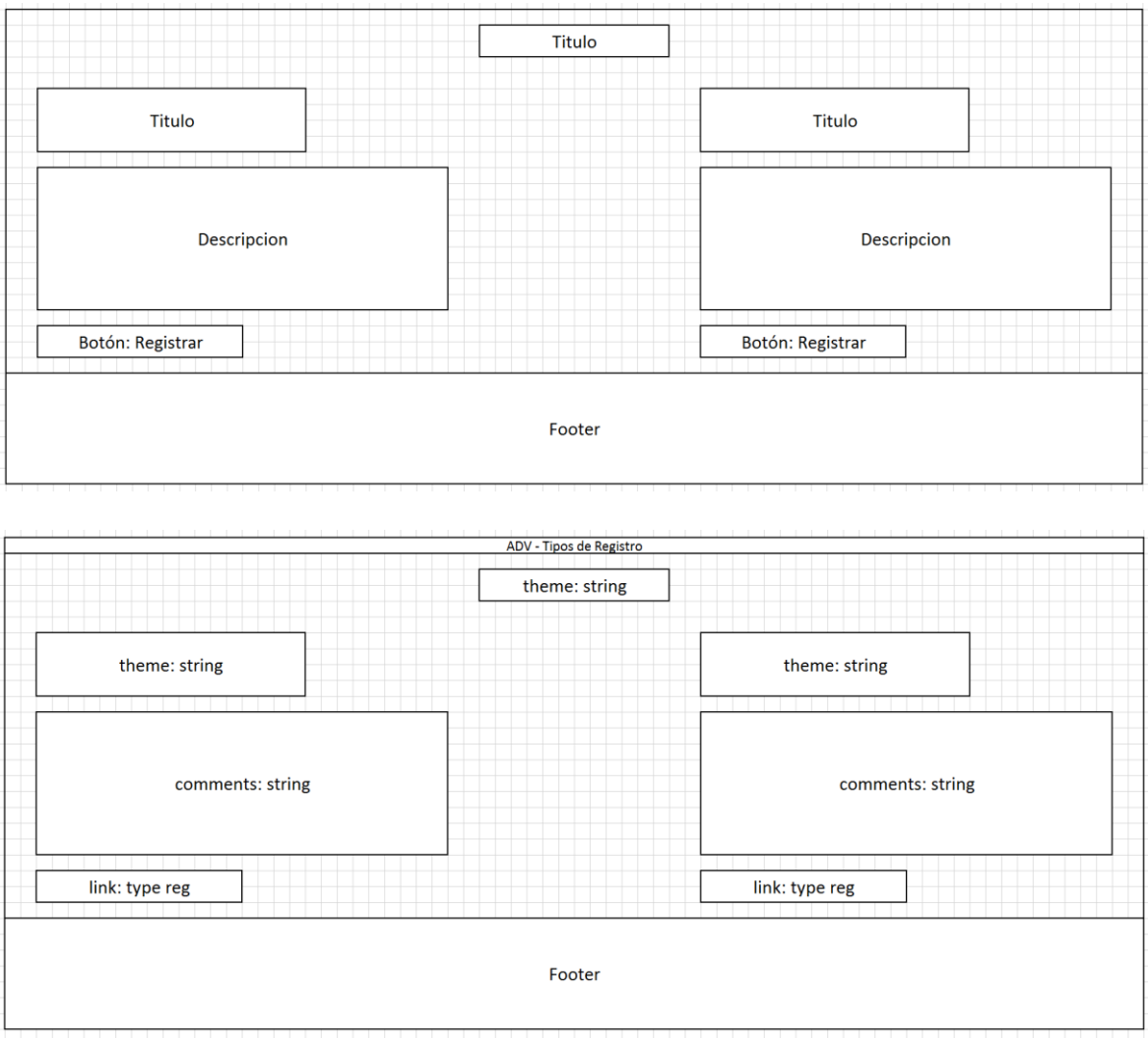


Fuente: Elaboración Propia

3.4.5. Modulo: Tipos de Registro

En este módulo los usuarios nuevos que deseen registrarse en la plataforma podrán seleccionar la categoría que deseen obtener ya sea que deseen buscar un servicio o que deseen promocionar uno.

Ilustración 32, ADV Modulo Tipos de Registro



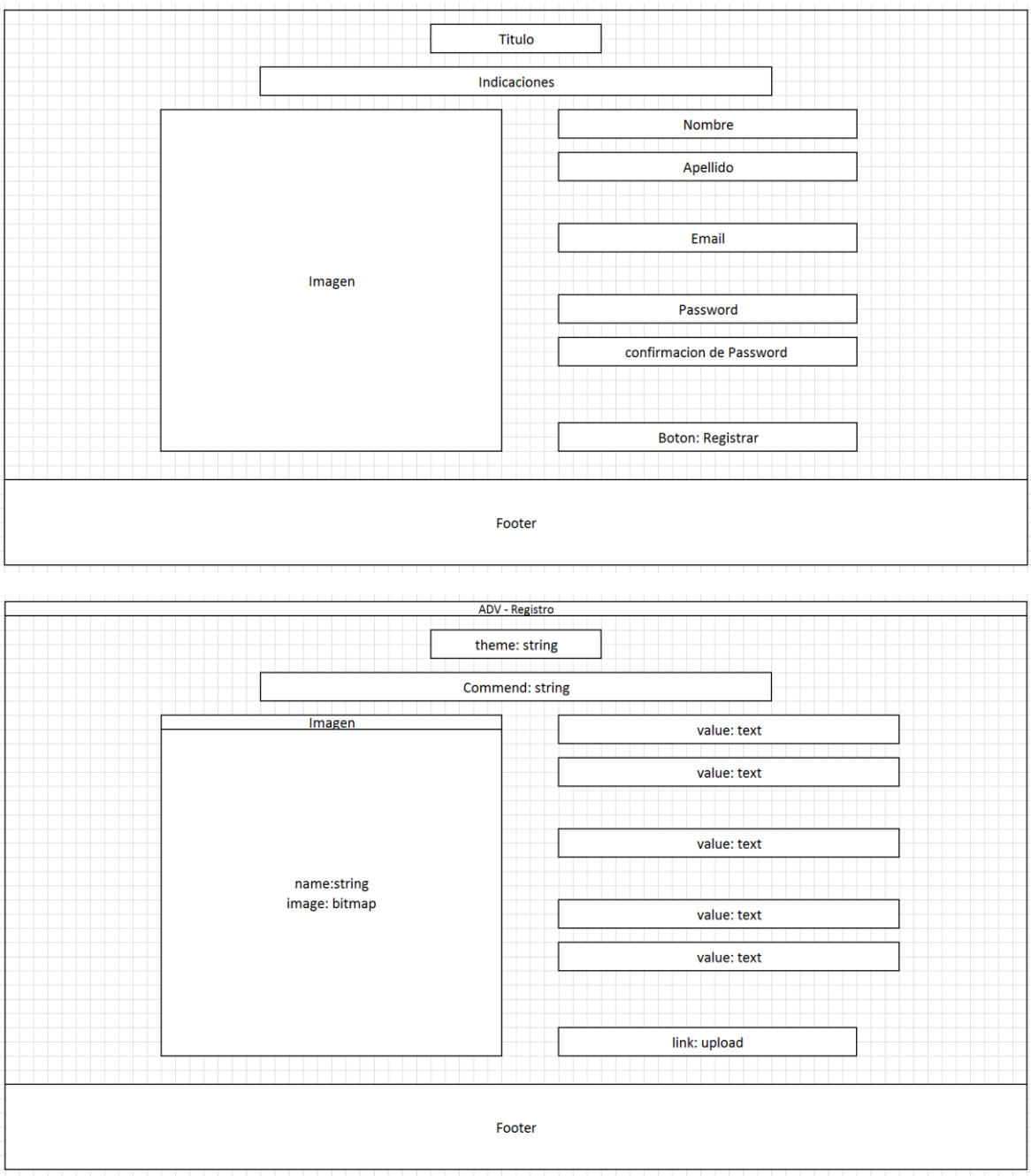
Fuente: Elaboración Propia

Cuenta con campos que contienen una descripción de las acciones que se le permitirá al usuario una vez se registre y dos botones que determinan el tipo de usuario que se desea registrar.

3.4.6. Modulo. Registro Ofertante / No ofertante

En este módulo los usuarios nuevos que deseen registrarse en la plataforma ya habrán seleccionado el tipo de registro lo que internamente les dará una categoría diferente sin embargo el módulo en el que se registraran compartirá características iniciales iguales, como son los campos de nombre, apellido, email/correo, contraseña y verificación de contraseña.

Ilustración 33, ADV Modulo Registro



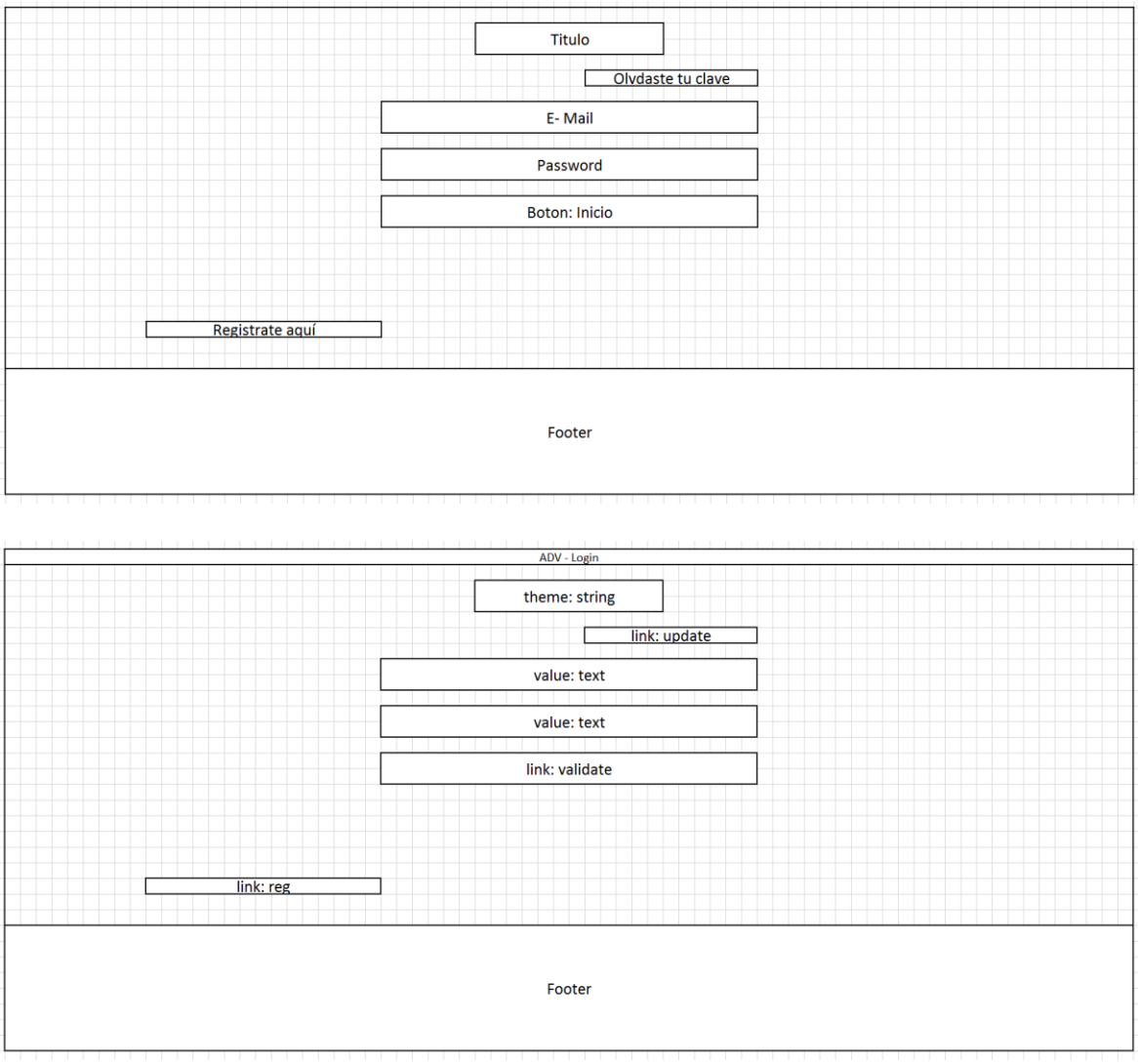
Fuente: Elaboración Propia

3.4.7. Modulo: Inicio de sección

El acceso previo al uso del total de las aplicaciones de la plataforma el cual está encargado de identificar qué tipo de usuario está accediendo y otorgarle los permisos necesarios, así como también, establecer las restricciones del caso.

Como módulo de logeo clásico este cuenta con dos campos, Email y contraseña con los cuales se precede a ser la verificación al dar clic en el botón que solicita el ingreso a la plataforma, además cuenta con dos textos vinculados con acciones como el registro en caso de no tener una cuenta activa o el restablecimiento de su contraseña en caso de no recordarla.

Ilustración 34, ADV Modulo Inicio de Sesión



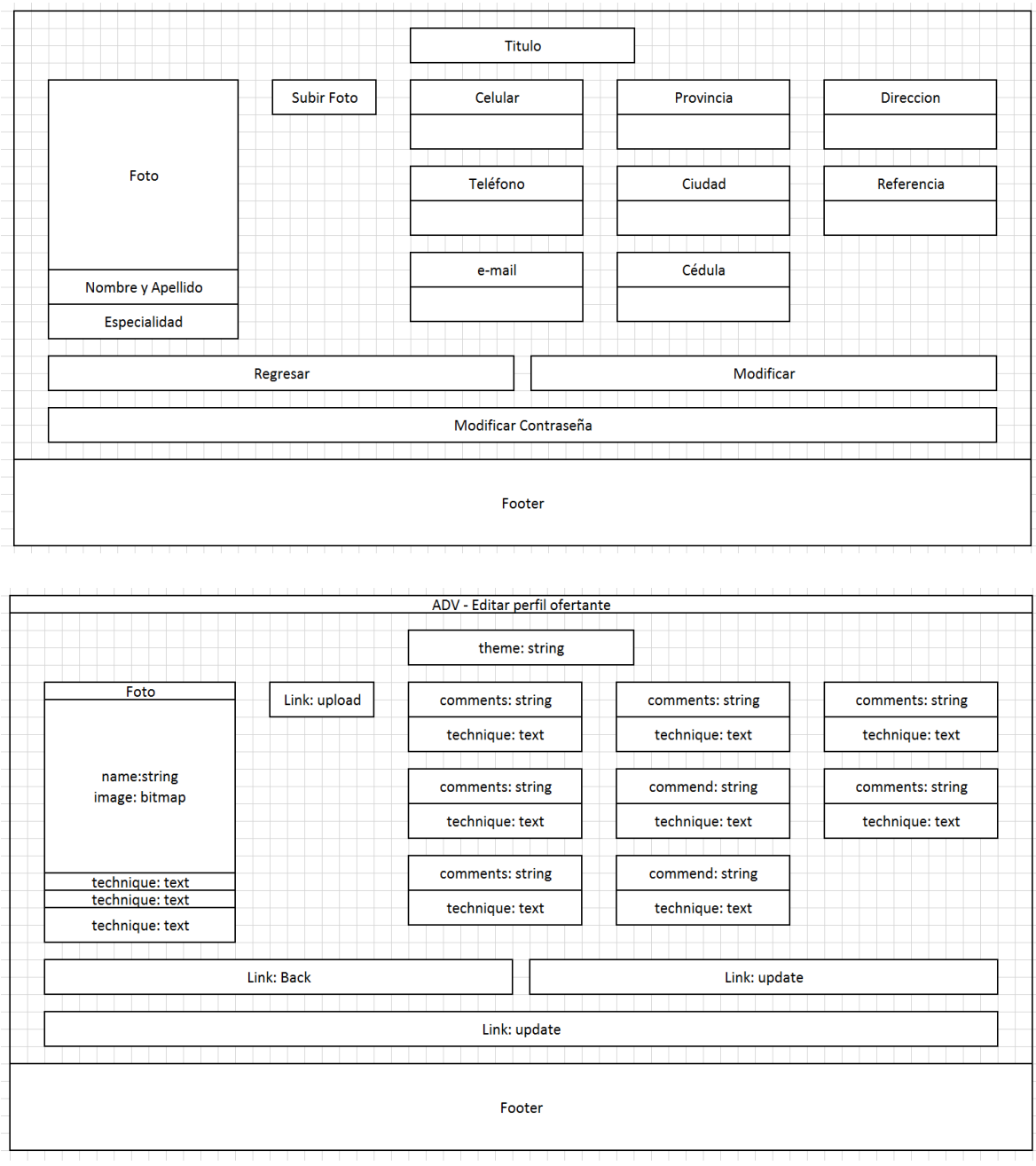
Fuente: Elaboración Propia

3.4.8. Modulo: Editar Perfil

El usuario puede registrar y modificar sus datos una vez este haya ingresado a la plataforma, si el usuario no completa el registro este no puede acceder a los métodos de búsqueda de servicios.

Perfil Ofertante

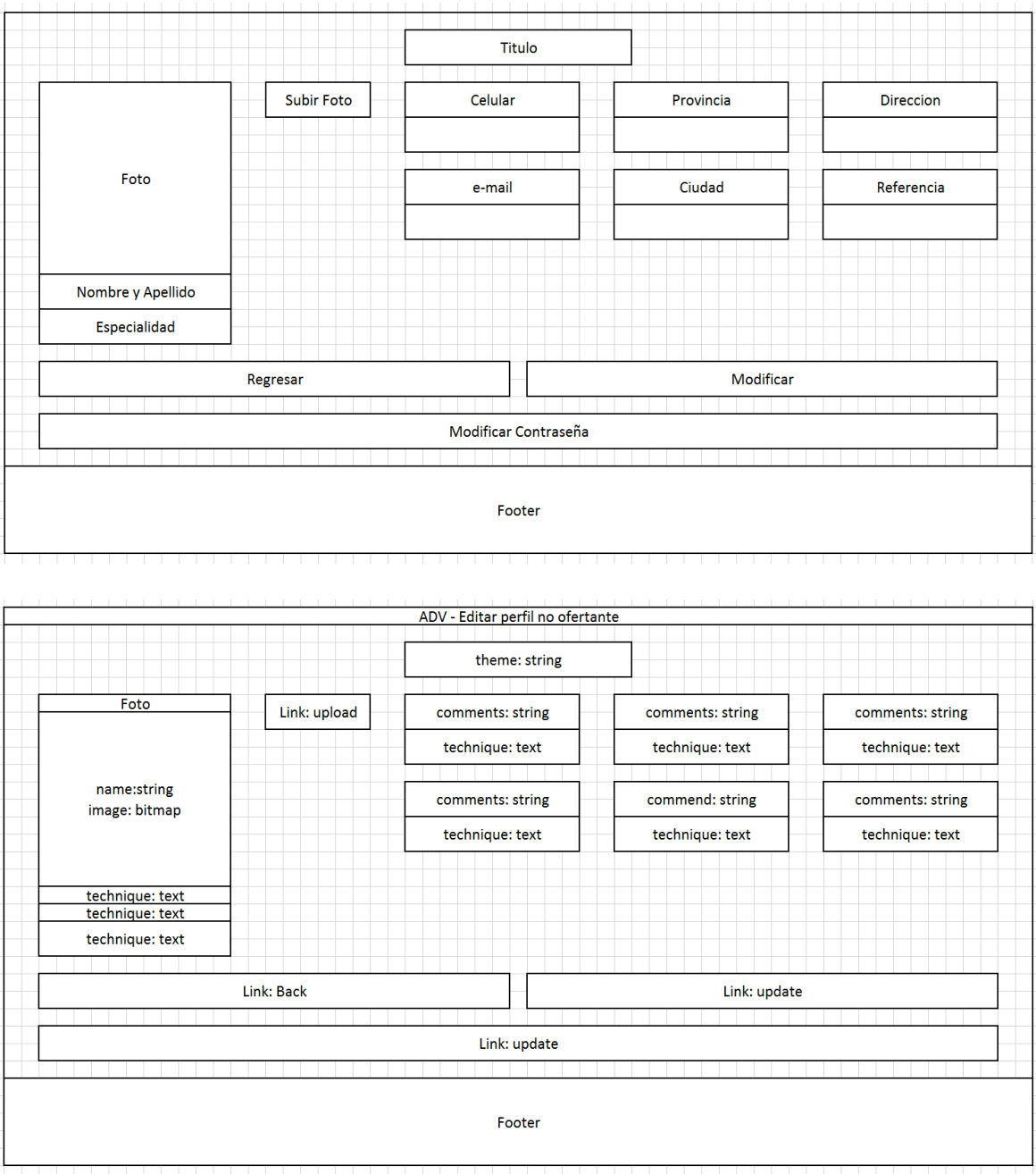
Ilustración 35, ADV Editar Perfil Ofertante



Fuente: Elaboración Propia

Perfil No ofertante

Ilustración 36, ADV Editar Perfil No Ofertante



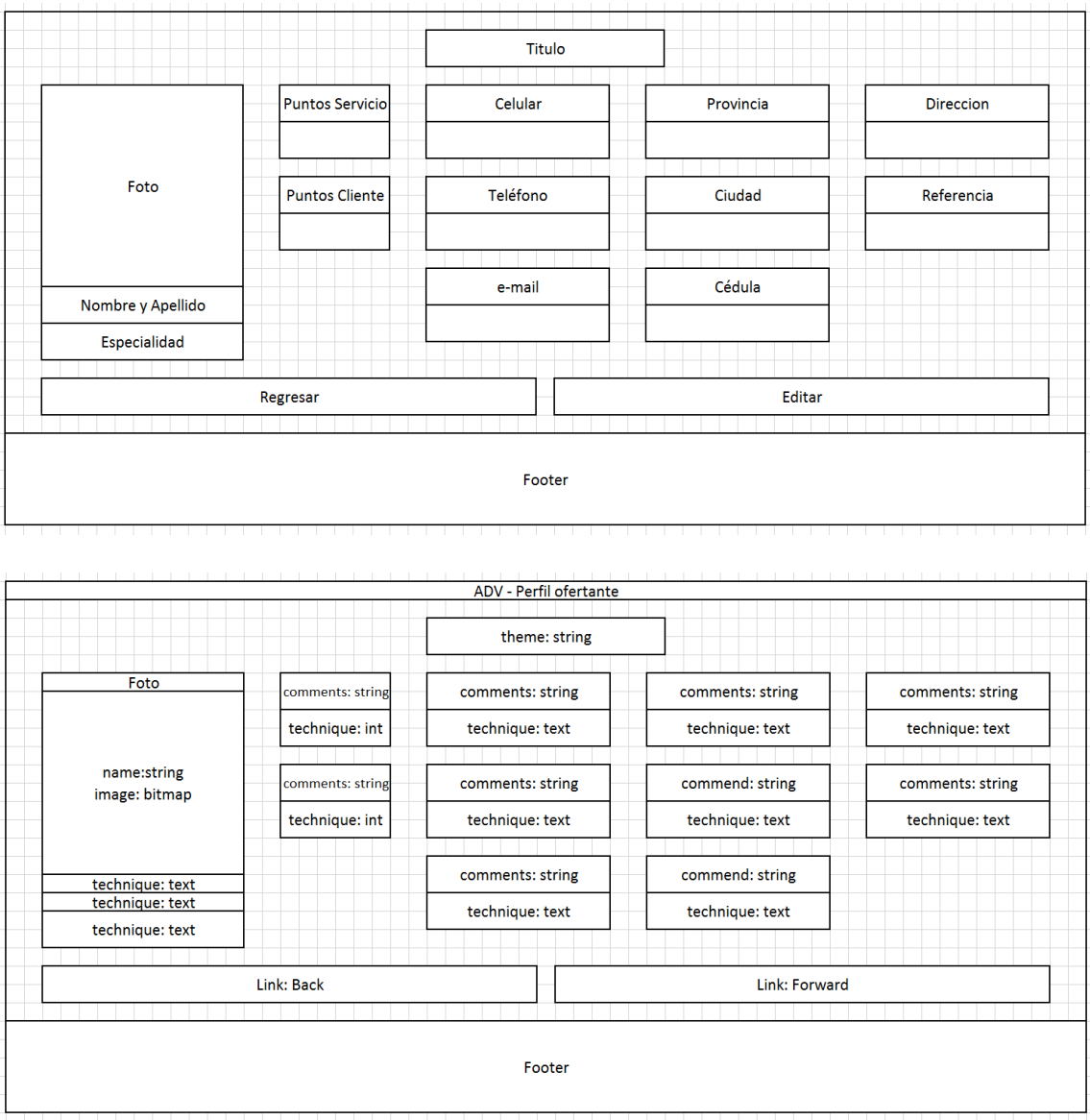
Fuente: Elaboración Propia

3.4.9. Modulo: Perfil de Ofertantes

El usuario ofertante de un servicio podrá visualizar su información ingresada previamente al completar el proceso de registro, así como acceder a la opción de modificarla la información de la plataforma ubicada en los botones de la sección inferior.

Los campos mostrados también incluyen los campos de calificación en donde se puede observar los puntos a los que se ha hecho acreedor el usuario ya sea por la calidad del servicio realizado o por la calidad de cliente al ser el usuario el que contrate un servicio.

Ilustración 37, ADV Perfil Ofertante



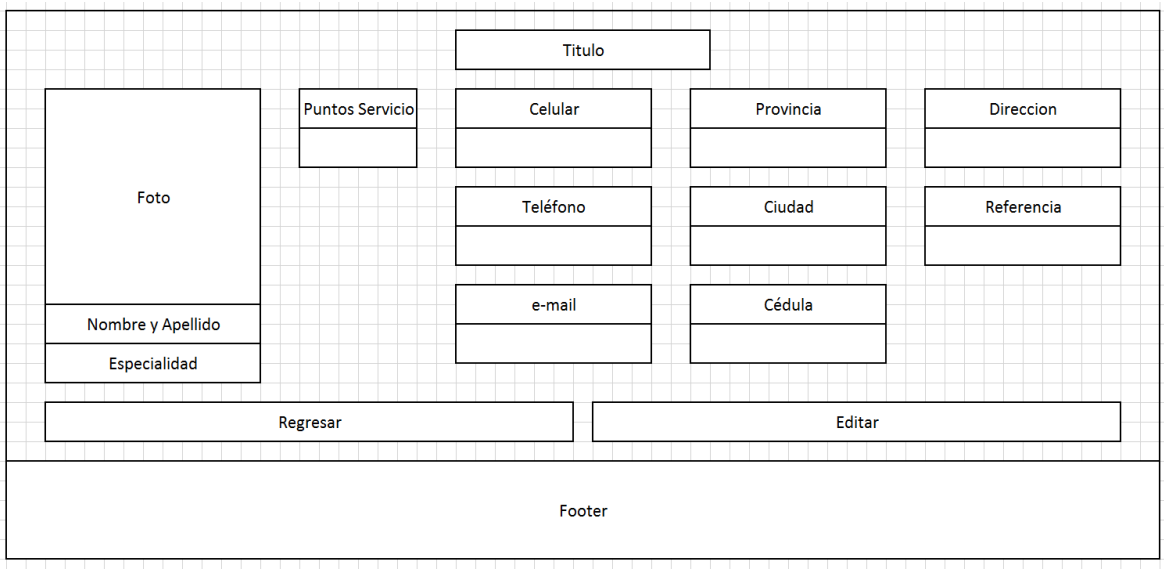
Fuente: Elaboración Propia

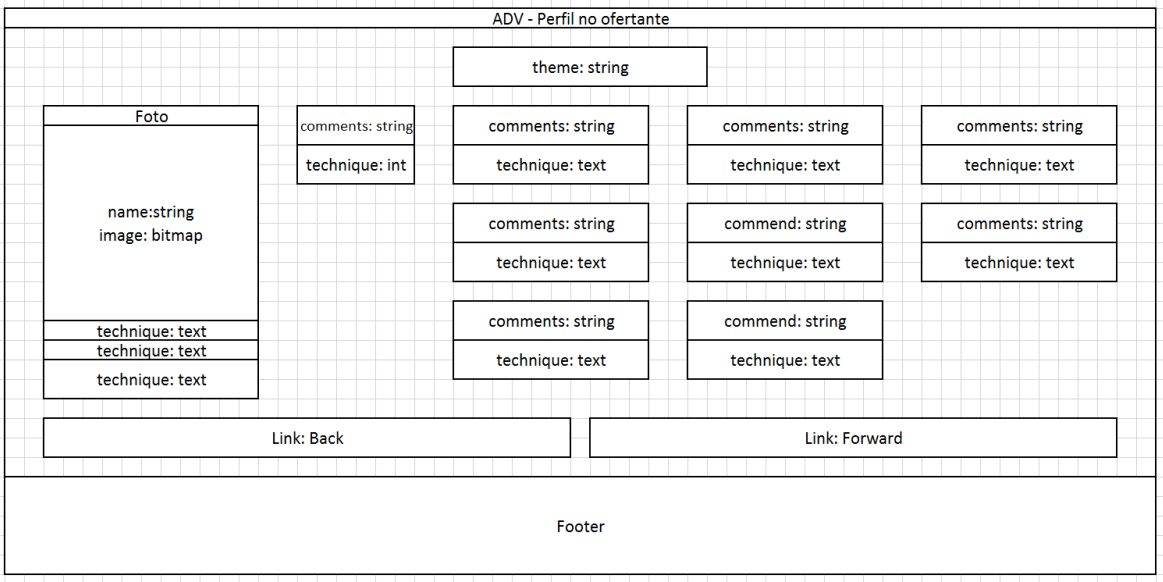
3.4.10. Modulo: Perfil de No ofertantes

El usuario no ofertante de un servicio puede visualizar su información, así como acceder a la opción de modificarla, o eliminar su cuenta de la plataforma.

Los campos mostrados también incluirán los campos de calificación en donde se puede observar los puntos a los que se ha hecho acreedor el usuario por su calidad de cliente al ser el usuario el que contrate un servicio.

Ilustración 38, ADV Perfil No Ofertante





Fuente: Elaboración Propia

3.4.11. Modulo: Bienvenido

El usuario que ha completado el proceso de registro y/o a iniciado sesión correctamente tiene acceso a la página de bienvenida, esta página cuenta con varias secciones con diferentes funciones, Las secciones con las que cuenta son:

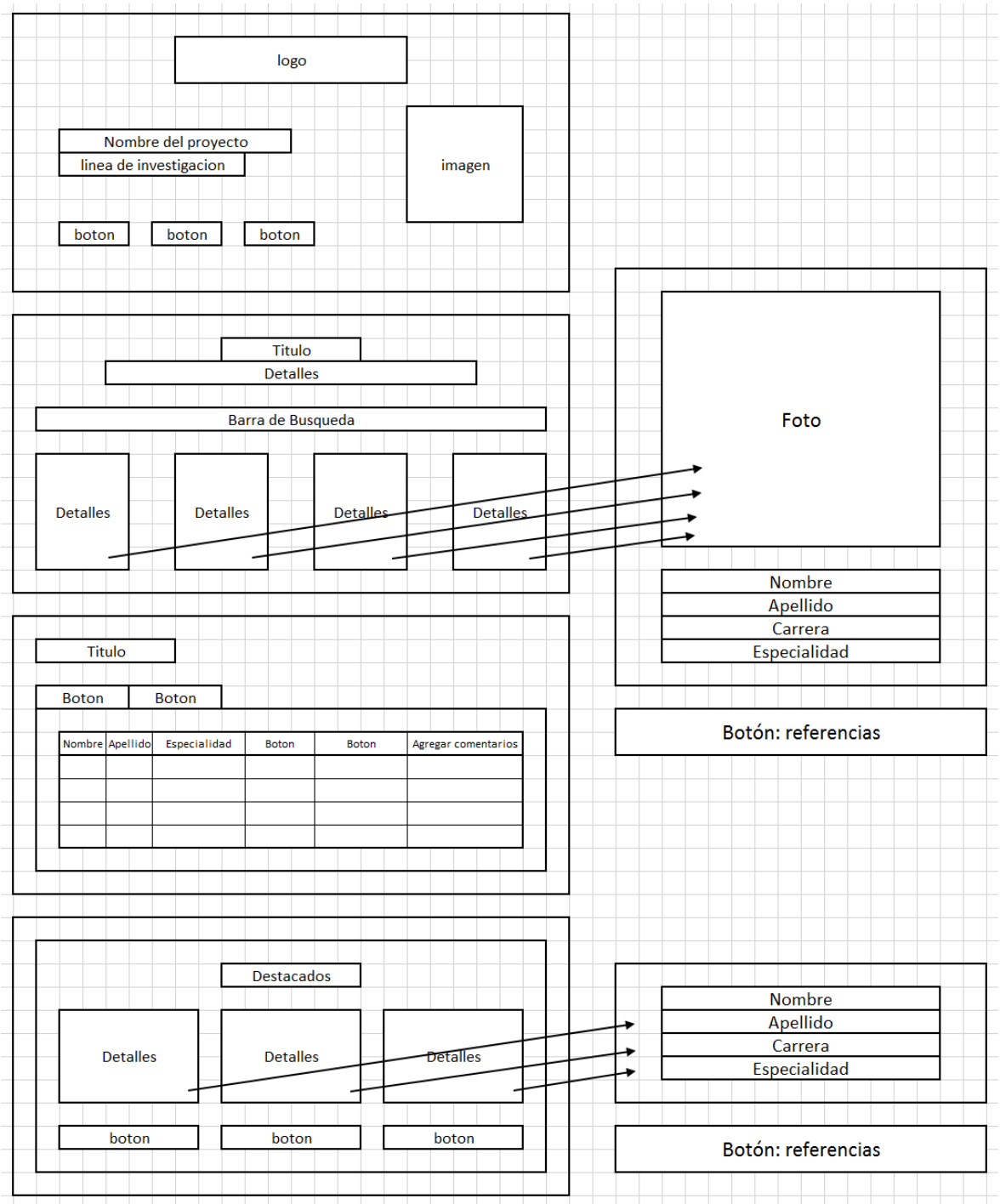
Cabecera con información del usuario como nombre, apellido y especialidad; así como también con accesos rápidos a la sección de búsqueda de servicios y pendientes, además de un botón de acceso al módulo de edición de perfil personal.

Sección de búsqueda rápida en donde accederá a los usuarios que se encuentran cerca de la ubicación registrada, así también el acceso a la búsqueda general donde observara a todos los usuarios ofertantes registrados.

Sección de pendientes donde se observa la información de las solicitudes pendientes de respuesta y aquellas que ya permiten el contacto y calificación de la atención del usuario al que se realizó la solicitud.

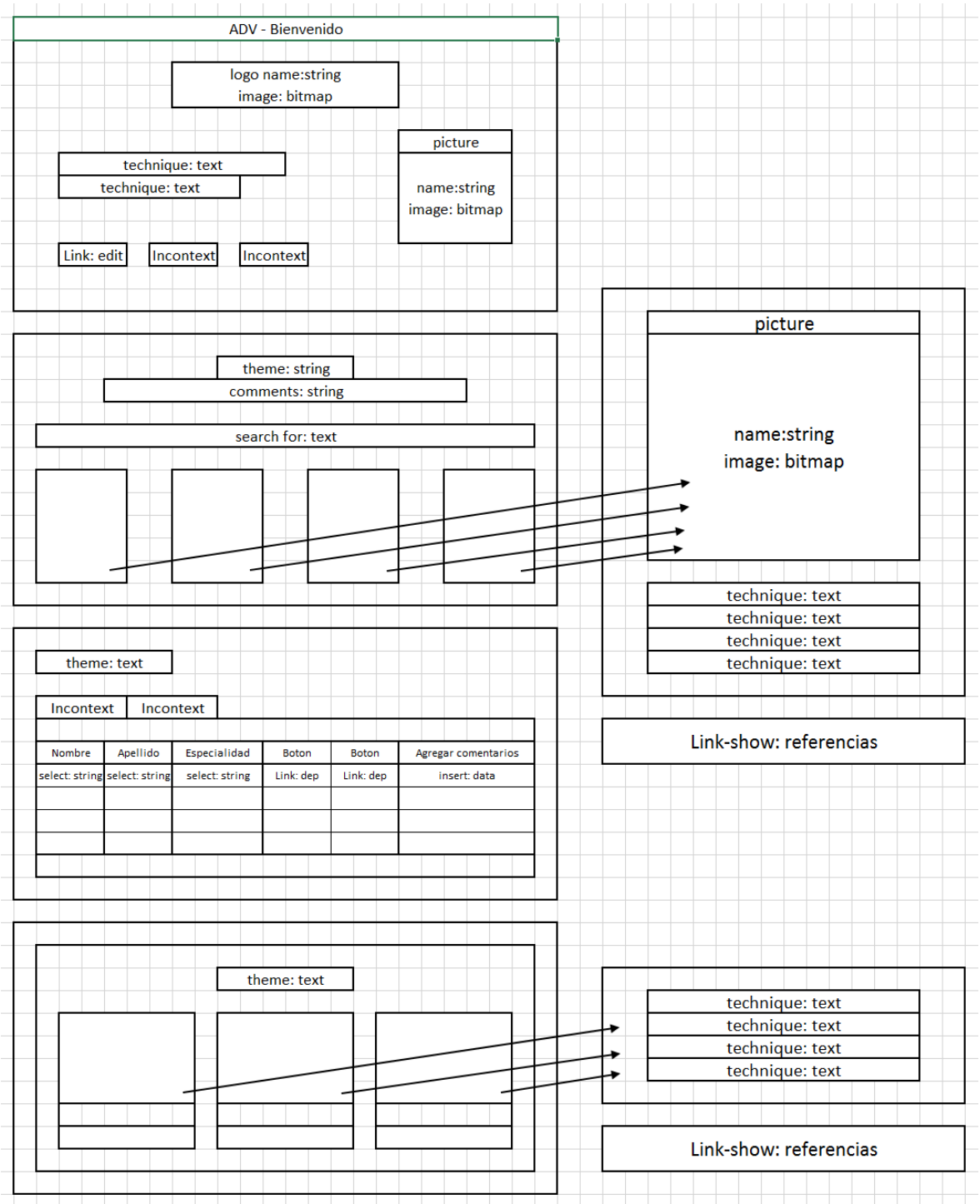
Por último, la sección de destacados en donde se observa la carrera, nombre, apellido y especialidad de los usuarios con el mejor puntaje en la plataforma para su fácil contratación.

Ilustración 39, ADV Modulo Bienvenido - Parte 1



Fuente: Elaboración Propia

Ilustración 40, ADV Modulo Bienvenido - Parte 2



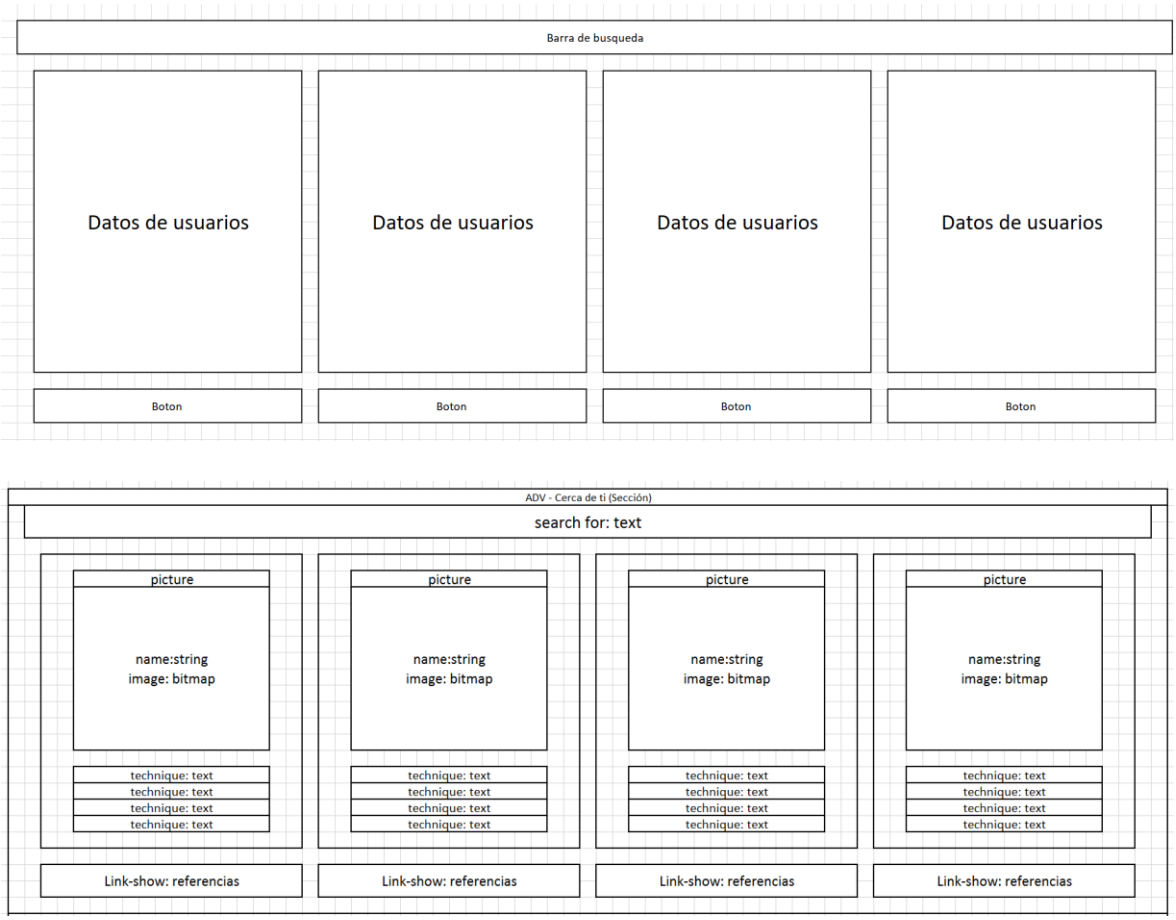
Fuente: Elaboración Propia

3.4.12. Sección: Cerca de ti

Esta sección proporciona un acceso rápido a los usuarios que ofertan sus servicios en la misma zona registrada por el usuario. Los usuarios de categoría ofertante no puede observar sus propios perfiles en la búsqueda.

Los datos del usuario que se mostrarán serán fotografía, nombre, apellido, carrera y especialidad.

Ilustración 41, ADV Sección - Cerca de ti

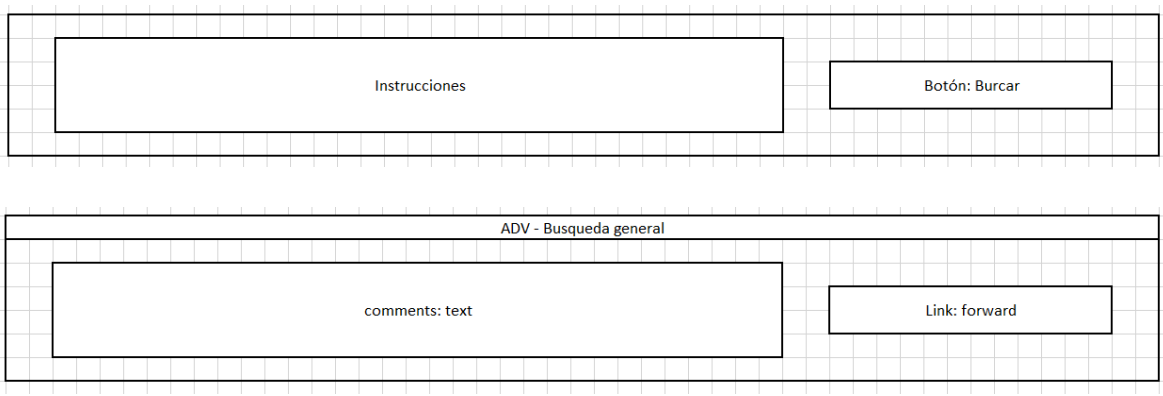


Fuente: Elaboración Propia

3.4.13. Sección: Búsqueda general

En esta sección los usuarios que no encuentren el servicio con las características que requieran en la sección cerca de ti podrán realizar una búsqueda más profunda en todos los usuarios registrados en la plataforma, estos detalles serán expuestos en el segmento proporcionado para información/instrucciones.

Ilustración 42, ADV Sección - Búsqueda General



Fuente: Elaboración Propia

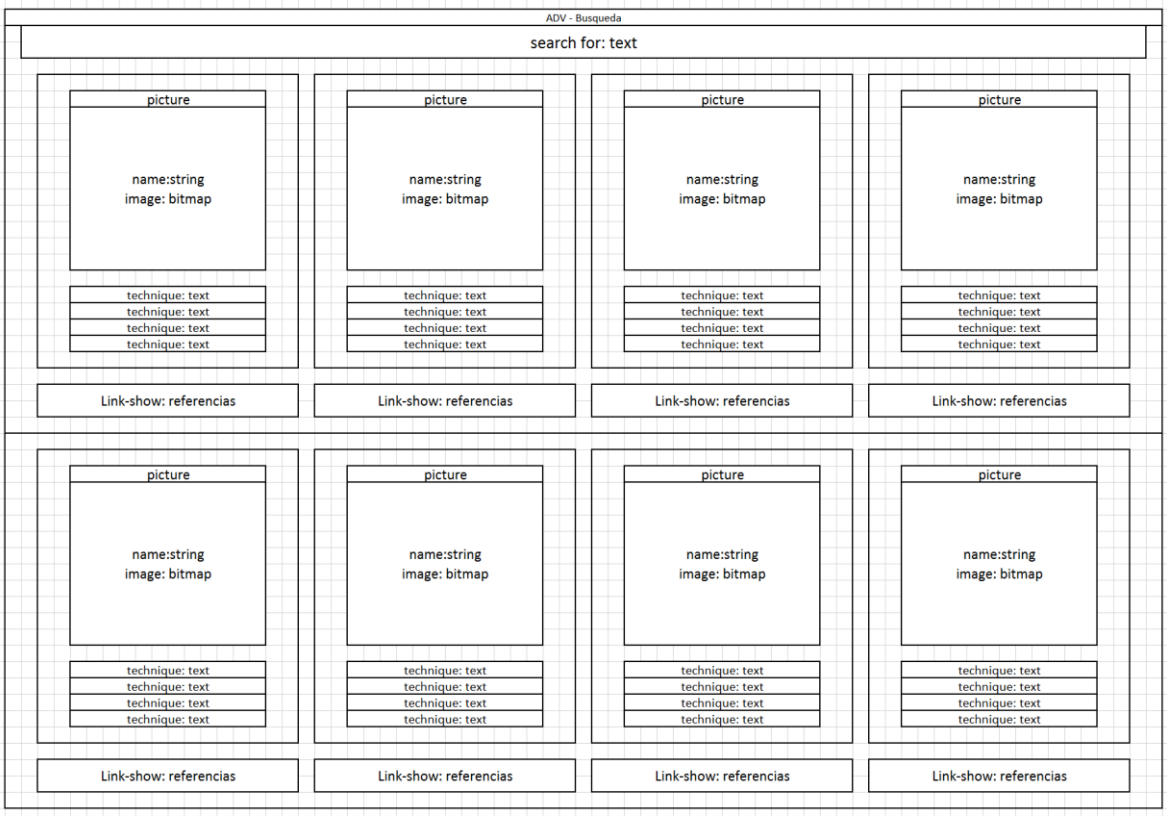
3.4.14. Módulo: Búsqueda general

En esta sección los usuarios que deseen contratar un servicio podrán acceder a toda la información de la base de datos permitida por el sistema, esto se realizara acorde a las restricciones de uso del usuario en cuestión, lo que quiere decir que, si el usuario no está registrado no puede acceder a la búsqueda, así como la información que obtenga de la plataforma es limitada.

En este módulo los datos del usuario que se mostrarán serán los mismos que en el módulo bienvenido sección cerca de ti, los datos revelados serán fotografía, nombre, apellido, carrera y especialidad.

Ilustración 43, Modulo Búsqueda General





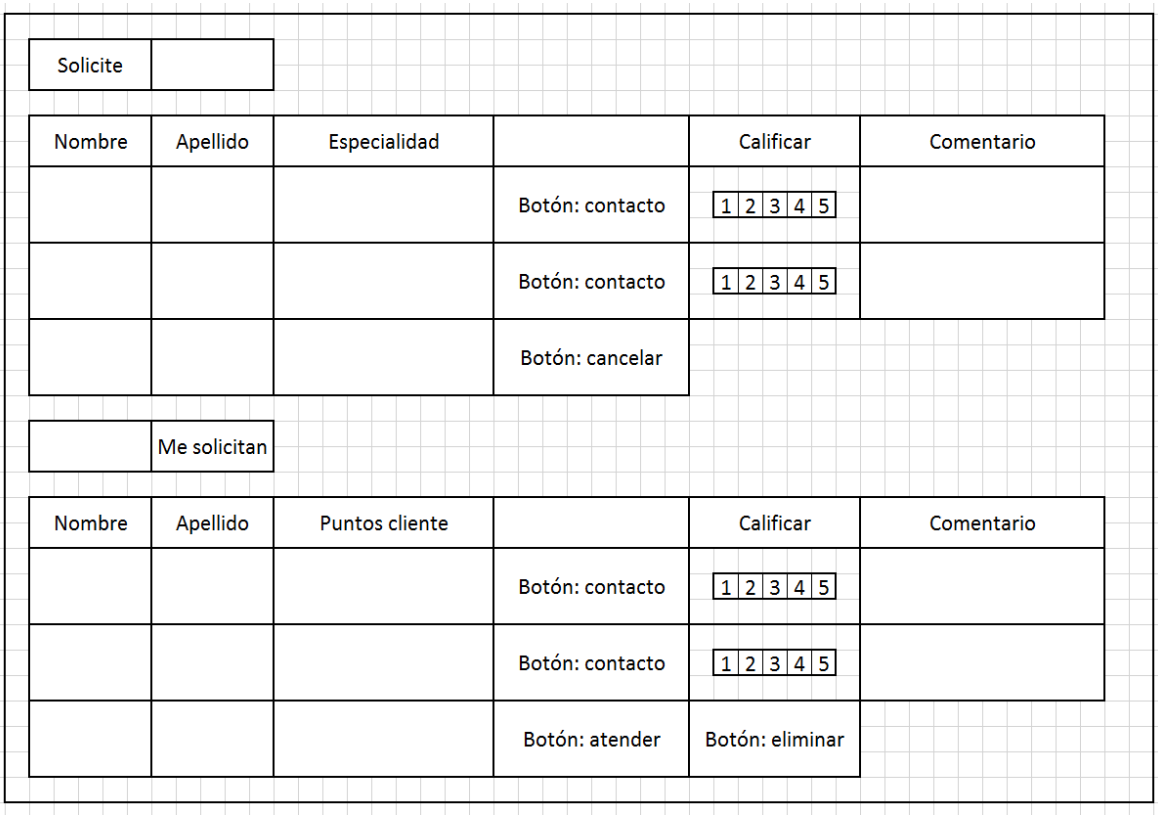
Fuente: Elaboración Propia

3.4.15. Sección: Pendientes

En esta sección los usuarios pueden revisar sus peticiones de servicio, así como calificar y comentar a los usuarios con los que ya haya trabajado; es necesario recalcar que los usuarios no administradores solo podrán tener acceso a la información detallada de sus contrapartes en la plataforma si dicha contraparte atiende la solicitud de servicio que se ha realizado.

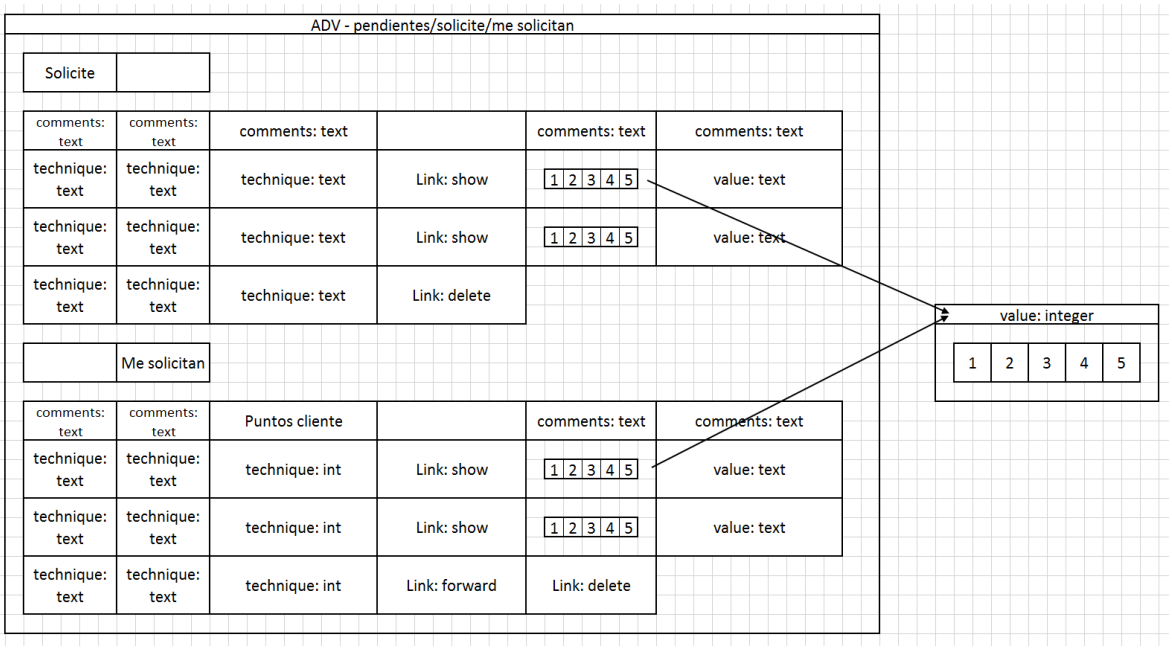
En esta misma sección, al atenderse una solicitud de servicio realizada por los usuarios no administradores se habilitará el campo de comentarios, la escala para calificación de 1 a 5 el cual se implementará por ser una escala de puntuación representativa en las aplicaciones *web*; por último, se mostrará el botón de contacto el que dará acceso a la información detallada de los usuarios.

Ilustración 44, ADV Sección Pendientes - Parte 1



Fuente: Elaboración Propia

Ilustración 45, ADV Sección Pendientes - Parte 2



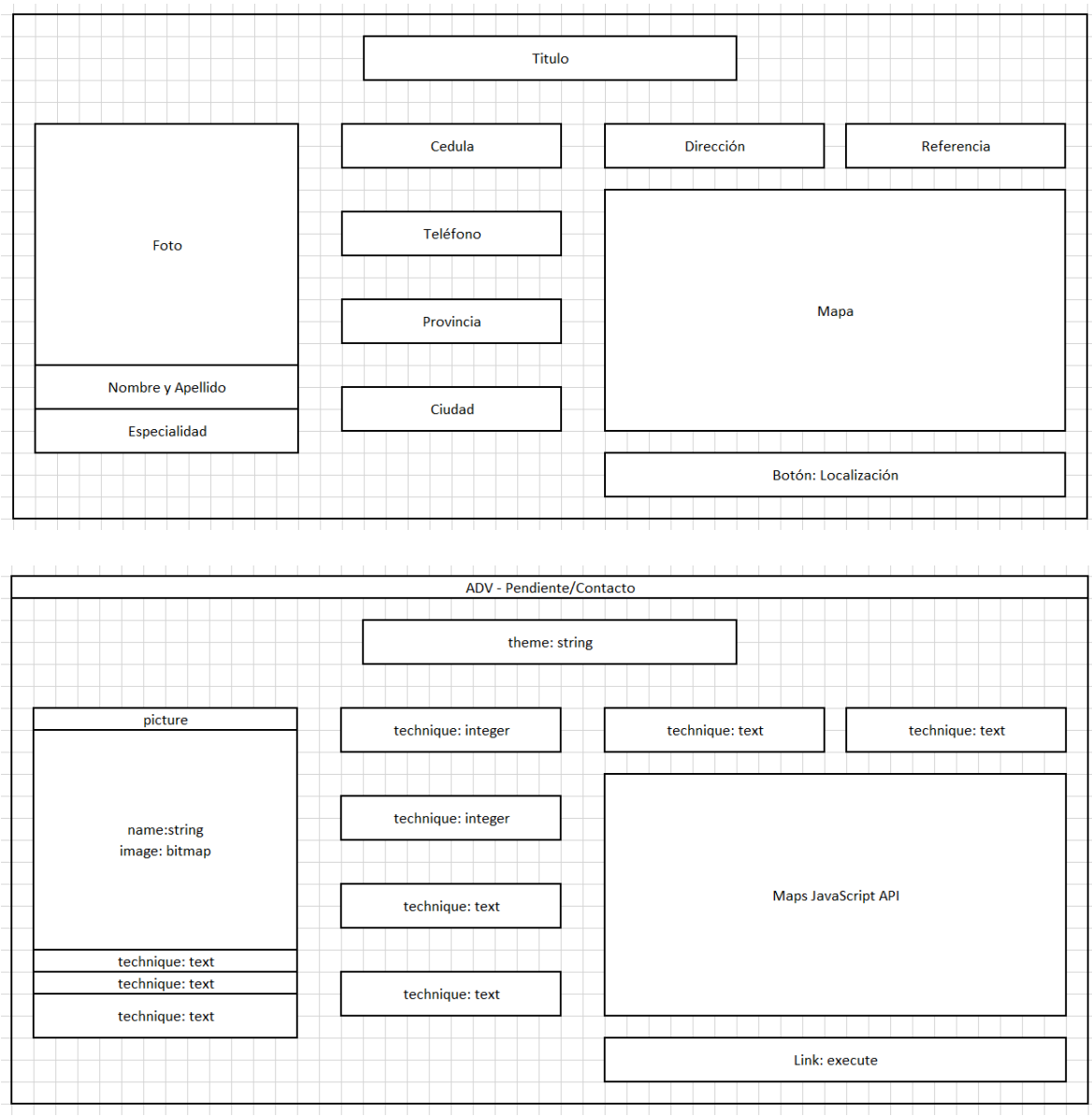
Fuente: Elaboración Propia

3.4.16. Sección: Pendientes / Botón de Contacto

Realizada la solicitud de servicio a un usuario ofertante si dicho usuario acepta la solicitud ambos usuarios podrán visualizar los datos de contacto y localización de su respectiva contraparte, la información mostrada de los usuarios será fotografía, nombre, apellido, carrera, especialidad, dirección, teléfonos y geolocalización.

La localización se realizará a través del mapa generado por *Google Maps* en conjunto con la codificación permitiendo a la plataforma determinar la localización del usuario en relación a los datos registrados por los usuarios.

Ilustración 46, ADV Sección Pendientes - Contacto

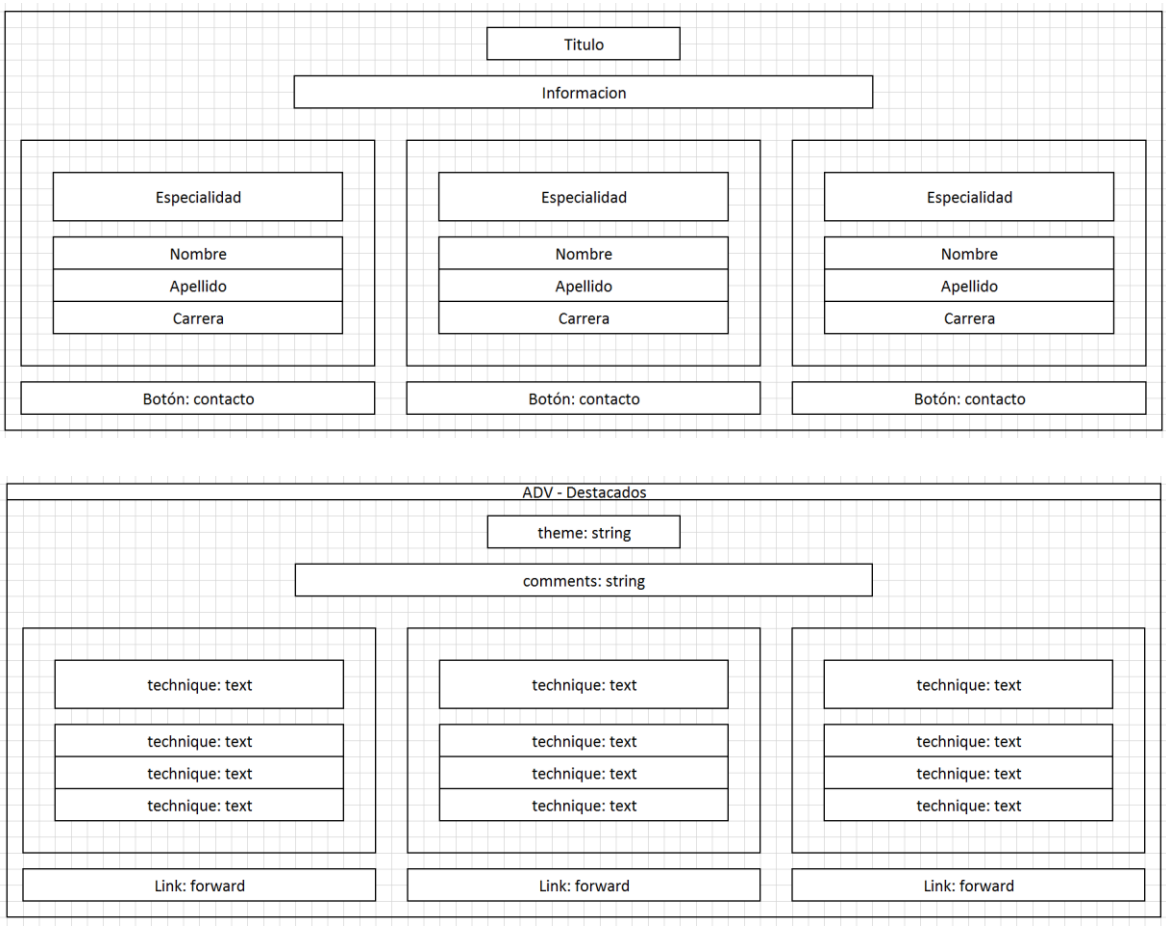


Fuente: Elaboración Propia

3.4.17. Sección: Destacados

En esta sección los usuarios no administradores con las mejores calificaciones en un rango de cuatro puntos siete a cinco tendrán una exposición inmediata sin necesidad de acceder a las secciones de búsqueda, mostrando sus datos en esta sección, los datos del usuario que se mostrarán serán nombre, apellido, carrera y especialidad.

Ilustración 47, ADV Sección Destacados



Fuente: Elaboración Propia

3.4.18. Modulo: Administración de usuarios

En este módulo el usuario administrador puede insertar y modificar registros de usuarios, consultar y eliminar usuarios no administradores de la plataforma, cada una de estas opciones podrán ser realizadas en este módulo únicamente por el administrador.

Ilustración 48, ADV Administración de Usuarios

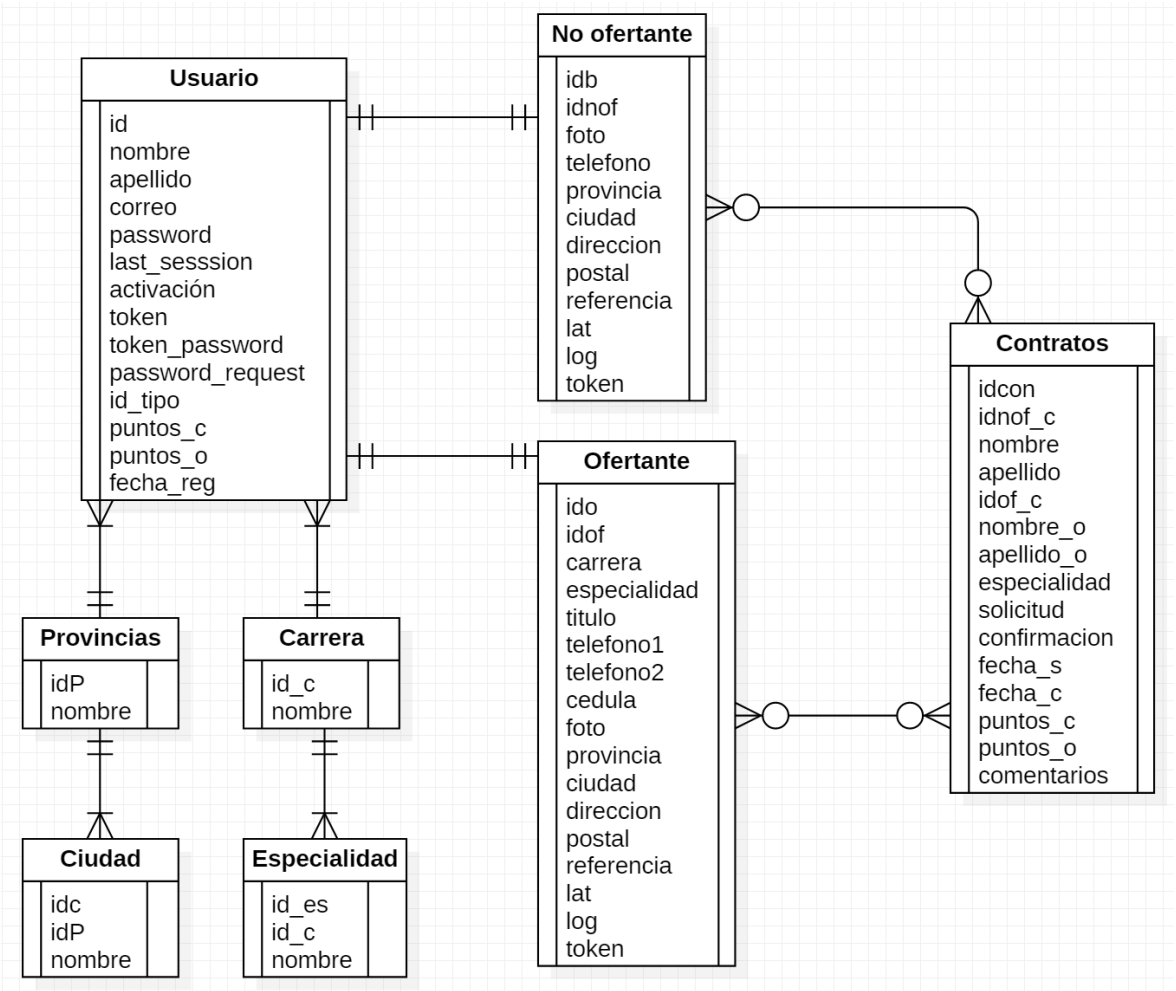


Fuente: Elaboración Propia

3.4.19. Modelo entidad / relación

Es el resultado de convertir los modelos conceptuales generados anteriormente en un modelo de datos lógico que se ajuste de forma correcta a la plataforma tanto en sus operaciones de consulta y actualización como en su funcionalidad al navegar en ella.

Ilustración 49, ERD Diagrama Entidad Relación



Fuente: Elaboración Propia

3.5. QUINTA ETAPA. IMPLEMENTACIÓN

En esta etapa todos los esquemas y teorías aplicadas en el documento son trasladadas a código real en cual está encargado de que se ejecuten las acciones dentro de la plataforma de manera correcta, en esta sesión y visualizando los anexos correspondientes se podrá constatar la codificación necesaría para el funcionamiento de la plataforma

3.5.1. Hosting y Dominio

La implementación de una plataforma *online* o *WebApp* exige necesariamente la adquisición de un *hosting* (servidor de alojamiento) y un dominio, los cuales cuentan con las características necesarias para brindar soporte a la plataforma; de tal forma que podemos resaltar detalles como son:

Almacenamiento HD de Espacio	800 MB
Transferencia mensual	20000 Mb
Servicios ASP	**Gold ASP**
Sever - Memoria (mínimo)	16GB
Velocidad de Servidores (mínimo)	6100 mhz
Correo Electrónico	**Gold ASP**
Alias	Ilimitado
Cuentas POP3 / FTP	Ilimitadas
Acceso vía *Web*	Sí
Auto - Contestadores	Ilimitadas
ASPeMail	Sí
Reenvió de Correo	Ilimitado
Registros MX editables	Sí
Otras Características	**Gold ASP**
Uso del FTP	Ilimitado
Cuentas FTP	Ilimitado
Sub-Dominios(ej.ventas.dominio.com)	Ilimitado

Protección con Contraseña	SÍ
Soporte vía HelpDesk 24/7	SÍ
Soporte en español	SÍ
Características Avanzadas	**Gold ASP**
ASP	SÍ
ASPUpload	SÍ
ASP.NET	SÍ
Access ODBC DSN´s	Ilimitado
SQL *Web* Managment	SÍ
MySQL Base de datos	Ilimitado
MSSQL 2012	SI - 300MB Espacio
MSXML 4	SÍ
PHP7	SÍ
CDO	SÍ
ASPJpeg	SÍ
Instalación 24 Horas	SÍ

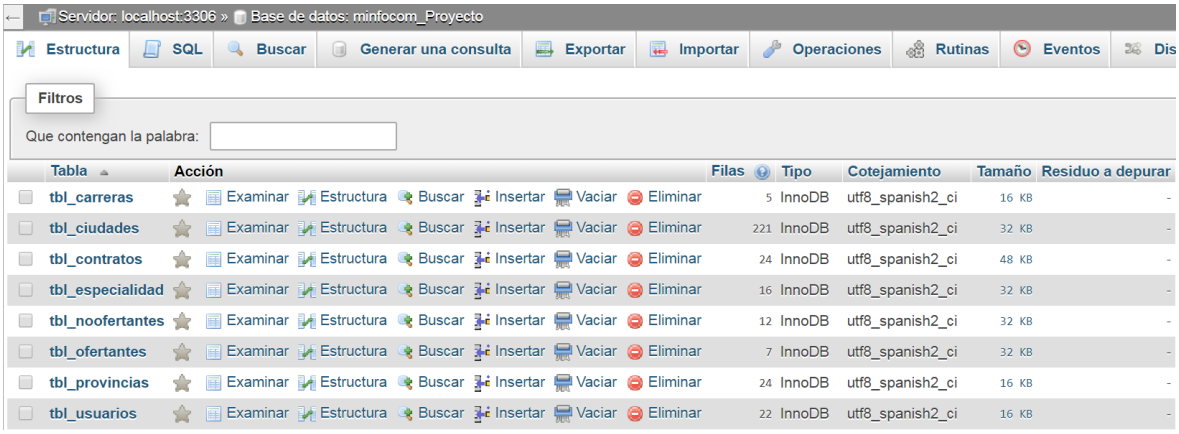
El dominio adquirido es www.minfo.com.ec el cual tiene categoría regional, esto no impide que sea visualizado o usado en otros países sin embargo denota que se creó principalmente para el mercado ecuatoriano.

3.5.2. Modelo lógico de la base de datos

En esta sección el único objetivo es trasladar la información del esquema lógico-abstracto al esquema lógico-físico, lo que se traduce en la implementación de métodos de acceso y almacenamiento para el control de las acciones dentro de la plataforma, en palabras más simples, es trasladar la información desarrollada en los pasos anteriores del método a un entorno de desarrollo, en este caso una base de datos digital.

La base de datos ha sido creada en un formato InnoDB con comprensión en lenguaje español latino UTF8-spanish2-ci, la base de datos trabaja en un servidor en línea en el cual se precedió a implementar la plataforma.

Ilustración 50, Modelo lógico BD



Fuente: Elaboración Propia

En las siguientes capturas se puede evidenciar la creación del sistema de información que almacena los datos generales y específicos de todos los usuarios de la plataforma.

Tabla – Usuarios

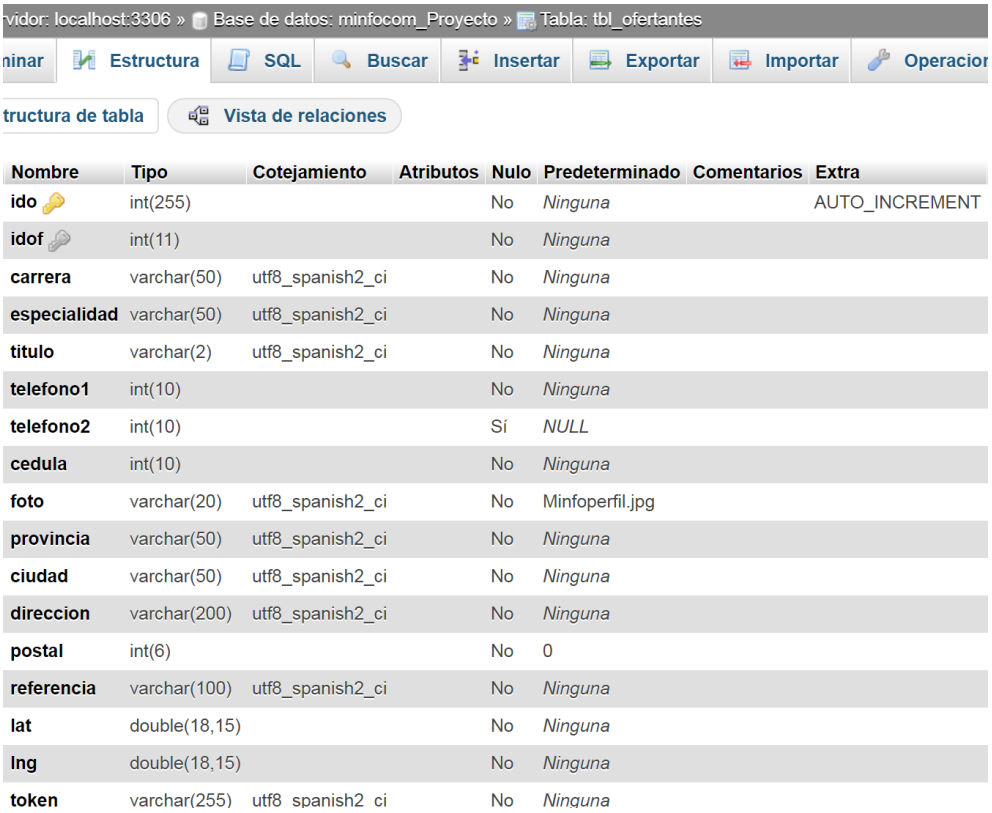
Ilustración 51, Tabla Usuarios BD

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
id	int(11)			No	*Ninguna*		AUTO_INCREMENT
nombre	varchar(50)	utf8_spanish2_ci		No	*Ninguna*		
apellido	varchar(50)	utf8_spanish2_ci		No	*Ninguna*		
correo	varchar(80)	utf8_spanish2_ci		No	*Ninguna*		
password	varchar(255)	utf8_spanish2_ci		No	*Ninguna*		
last_session	datetime			Sí	*NULL*		
activacion	int(11)			No	0		
token	varchar(255)	utf8_spanish2_ci		No	*Ninguna*		
token_password	varchar(100)	utf8_spanish2_ci		Sí	*NULL*		
password_request	int(11)			Sí	0		
id_tipo	varchar(50)	utf8_spanish2_ci		No	*Ninguna*		
puntos_c	float(3,2)			No	0.00		
puntos_o	float(3,2)			No	0.00		
fecha_reg	datetime			Sí	*NULL*		

Fuente: Elaboración Propia

Tabla – Ofertantes

Ilustración 52, Tabla Ofertantes BD,



Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
ido	int(255)			No	*Ninguna*		AUTO_INCREMENT
idof	int(11)			No	*Ninguna*		
carrera	varchar(50)	utf8_spanish2_ci		No	*Ninguna*		
especialidad	varchar(50)	utf8_spanish2_ci		No	*Ninguna*		
titulo	varchar(2)	utf8_spanish2_ci		No	*Ninguna*		
telefono1	int(10)			No	*Ninguna*		
telefono2	int(10)			Sí	*NULL*		
cedula	int(10)			No	*Ninguna*		
foto	varchar(20)	utf8_spanish2_ci		No	Minfoperfil.jpg		
provincia	varchar(50)	utf8_spanish2_ci		No	*Ninguna*		
ciudad	varchar(50)	utf8_spanish2_ci		No	*Ninguna*		
direccion	varchar(200)	utf8_spanish2_ci		No	*Ninguna*		
postal	int(6)			No	0		
referencia	varchar(100)	utf8_spanish2_ci		No	*Ninguna*		
lat	double(18,15)			No	*Ninguna*		
lng	double(18,15)			No	*Ninguna*		
token	varchar(255)	utf8_spanish2_ci		No	*Ninguna*		

Fuente: Elaboración Propia

Tabla - No ofertantes

Ilustración 53, Tabla No Ofertantes BD

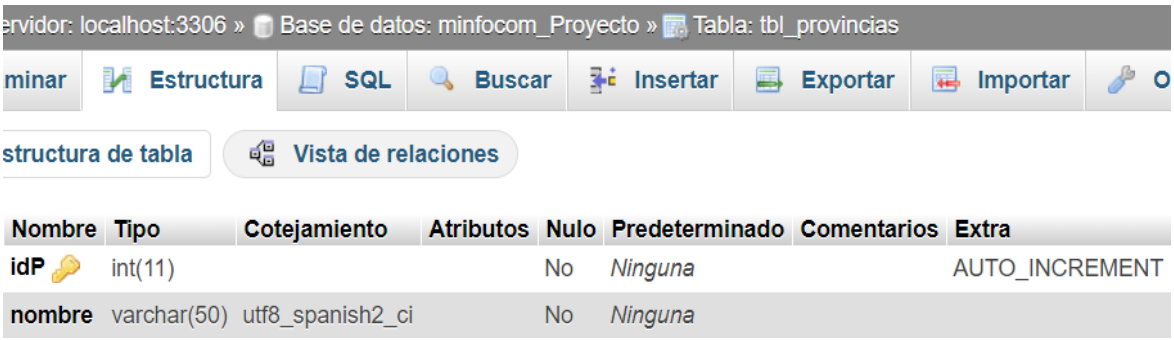


Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
idb	int(11)			No	*Ninguna*		AUTO_INCREMENT
idnof	int(11)			No	*Ninguna*		
foto	varchar(20)	utf8_spanish2_ci		No	Minfoperfil.jpg		
telefono1	int(10)			No	*Ninguna*		
provincia	varchar(50)	utf8_spanish2_ci		No	*Ninguna*		
ciudad	varchar(50)	utf8_spanish2_ci		No	*Ninguna*		
direccion	varchar(100)	utf8_spanish2_ci		No	*Ninguna*		
postal	int(6)			No	0		
referencia	varchar(100)	utf8_spanish2_ci		No	*Ninguna*		
lat	double			No	0		
lng	double			No	0		
token	varchar(255)	utf8_spanish2_ci		No	no ofertante		

Fuente: Elaboración Propia

Tabla - Provincias

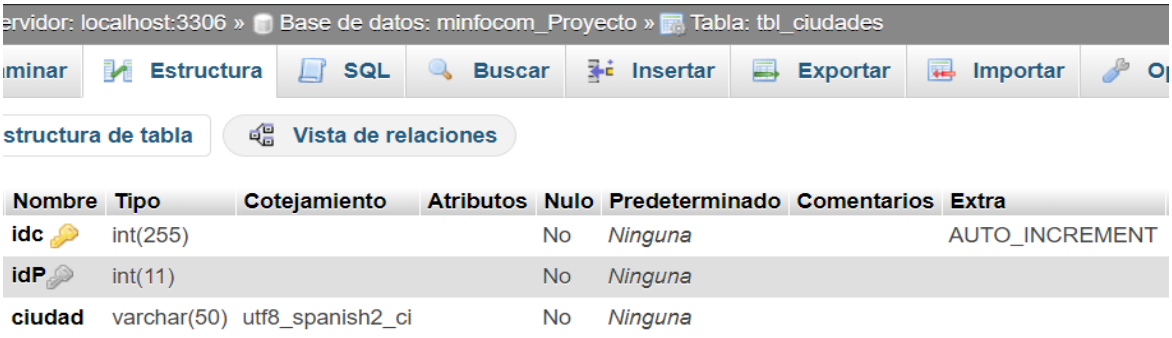
Ilustración 54, Tabla Provincias BD,



Fuente: Elaboración Propia

Tabla - Ciudades

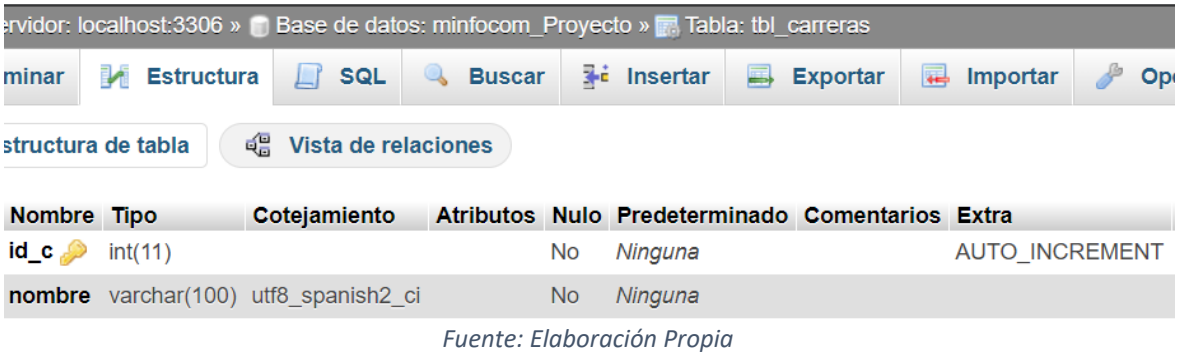
Ilustración 55, Tabla Ciudades BD,



Fuente: Elaboración Propia

Tabla - Carreras

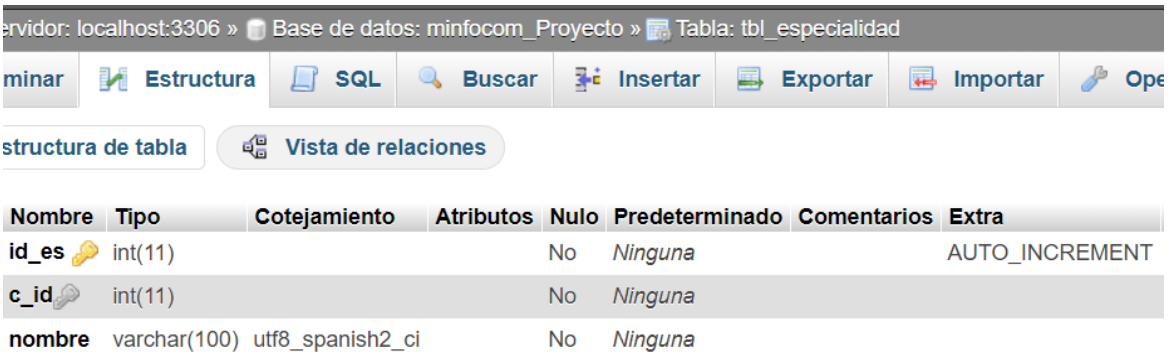
Ilustración 56, Tabla Carreras BD



Fuente: Elaboración Propia

Tabla – Especialidad

Ilustración 57, Tabla Especialidades BD,



Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
id_es	int(11)			No	*Ninguna*		AUTO_INCREMENT
c_id	int(11)			No	*Ninguna*		
nombre	varchar(100)	utf8_spanish2_ci		No	*Ninguna*		

Fuente: Elaboración Propia

Tabla – Contratos

Debe llamar la atención la similitud de los campos nombre con nombre_o y apellido con apellido_o sin embargo al ser datos heredados de la tabla usuario de los dos diferentes tipos de usuarios almacenaran el mismo tipo de dato, pero de diferentes usuarios.

Ilustración 58, Tabla Contratos BD



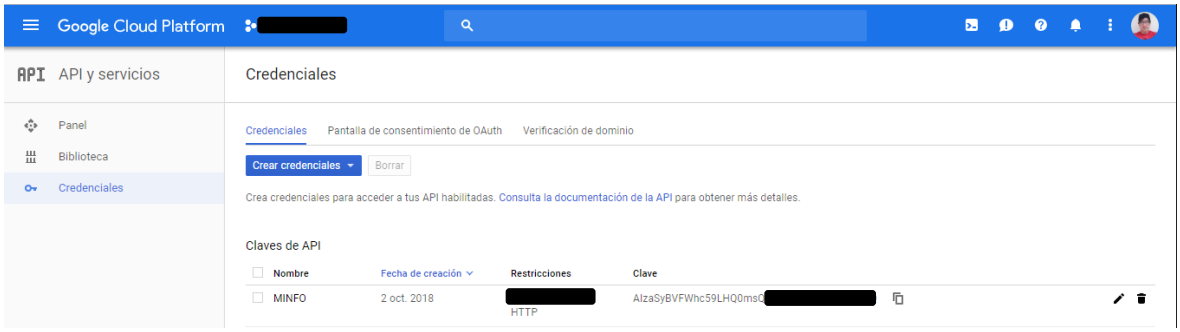
Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
idcon	int(11)			No	*Ninguna*		AUTO_INCREMENT
idnof_c	int(11)			No	*Ninguna*		
nombre	varchar(50)	utf8_spanish2_ci		No	*Ninguna*		
apellido	varchar(50)	utf8_spanish2_ci		No	*Ninguna*		
idof_c	int(11)			No	*Ninguna*		
nombre_o	varchar(50)	utf8_spanish2_ci		No	*Ninguna*		
apellido_o	varchar(50)	utf8_spanish2_ci		No	*Ninguna*		
especialidad	varchar(50)	utf8_spanish2_ci		No	*Ninguna*		
solicitud	int(1)			No	0		
confirmacion	int(1)			No	0		
fecha_s	datetime			Sí	*NULL*		
fecha_c	datetime			Sí	*NULL*		
puntos_c	int(1)			No	0		
puntos_o	int(1)			No	0		
comentarios	varchar(255)	utf8_spanish2_ci		No	Sin comentario		

Fuente: Elaboración Propia

3.5.3. Geolocalización

El proyecto cuenta con un *API KEY* específico para la plataforma que permite el uso de las herramientas de plataforma de desarrollo de *Google Maps* en este caso *Maps JavaScript API* principalmente, las cuales permiten que usemos sus librerías para acceder a diferentes funciones como son la geolocalización usando la base de datos de Google, la geo-codificación de datos que permite extraer coordenadas y características adicionales como altura y referencias de una ubicación específica, todas estas administradas a través de la codificación de JavaScript prediseñada por Google [38].

Ilustración 59, API KEY Google



Fuente: Elaboración Propia

A continuación, se expone cómo funciona la implementación de geolocalización al explicar los diferentes puntos del código usado.

Función de geolocalización de usuario en el módulo de registro final – *Maps JavaScript API*

Se inicia creando la función principal

```
function initMap ()
{
```

A continuación, se usa *Google Maps API* para geolocalizar el usuario a través de las funciones de ubicación del navegador usado por usuario de la plataforma.

```
navigator.geolocation.getCurrentPosition(
function (position){
coords =  {
   lng: position.coords.longitude,
   lat: position.coords.latitude
};
setMapa(coords);
```

Se envían las coordenadas al método para la creación del mapa.

```
},function(error){console.log(error);});
}
function setMapa (coords)
{
```

Se crea una nueva instancia del objeto mapa.

```
var map = new google.maps.Map(document.getElementById('map'),
{
 zoom: 15,
 center:new google.maps.LatLng(coords.lat,coords.lng),

});
```

Se crea el marcador en el mapa con sus respectivas propiedades, se añade el atributo *draggable* en true y en position se pondrá las mismas coordenadas que se obtuvieron en la geolocalización.

```
marker = new google.maps.Marker({
 map: map,
 draggable: true,
 animation: google.maps.Animation.DROP,
 position: new google.maps.LatLng(coords.lat,coords.lng),

});
```

Se agrega un nuevo evento al marcador y se adiciona la función *callback* igual que el evento *dragend* que indica que el usuario ha colocado el marcador sobre el mapa.

```
marker.addListener('click', toggleBounce);

marker.addListener( 'dragend', function (event)
{
```

Se almacenan las coordenadas de la posición del marcador en el input #coords.

```
    document.getElementById("lat").value = this.getPosition().lat();
    document.getElementById("lng").value = this.getPosition().lng();
  });

  var geocoder = new google.maps.Geocoder();
```

Se asigna la función del evento *dragend* del marcado anterior.

```
  google.maps.event.addListener(marker, 'dragend', function() {
  geocoder.geocode({'latLng': marker.getPosition()}, function(results, status) {
  if (status == google.maps.GeocoderStatus.OK) {
  var address=results[0]['formatted_address'];

  document.getElementById("direccion1").value= address;

  }
  });
  });

}
```

Se genera un *callback* al hacer clic en el marcador que retira o coloca la animación *BOUNCE*

```
function toggleBounce() {
  if (marker.getAnimation() !== null) {
    marker.setAnimation(null);
  } else {
    marker.setAnimation(google.maps.Animation.BOUNCE);
  }
}
```

Función de geolocalización de usuario en la sección pendientes botón contacto - *Maps JavaScript API*

Se inicia con la creación de un objeto mapa y especificamos el elemento DOM en el cual se muestra.

```
mapa.map = new google.maps.Map(document.getElementById('mapa'), {
```

También se especifican las funciones adicionales que se pondrán disponibles dentro del elemento.

```
  scrollwheel: false,
  zoom: 14,
  zoomControl: true,
  rotateControl : false,
  mapTypeControl: true,
  streetViewControl: false,
 });
```

Se crea el marcador inicial que aparecerá en el mapa directamente al ejecutarse el módulo de la plataforma como una dirección de latitud y longitud especifica:

```
mapa.marker = new google.maps.Marker({
 position: {lat: -1.254569, lng: -78.623851},
 draggable: true
 });
```

O aplicando la localización automática con las funciones de ubicación del dispositivo desde el que se accede a la plataforma.

```
if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(function(position) {
          var pos = {
            lat: position.coords.latitude,
            lng: position.coords.longitude
          };

          infoWindow.setPosition(pos);
          infoWindow.setContent('Location found.');
          infoWindow.open(map);
          map.setCenter(pos);
        }, function() {
          handleLocationError(true, infoWindow, map.getCenter());
        });
      } else {
```

En caso de no ser soportada la geolocalización por el navegador implementamos el código que informa del error de habilitación de localización.

```
        handleLocationError(false, infoWindow, map.getCenter());
      }
```

```
    }
```

Se asignan los marcadores al mapa destinado.

```
 mapa.marker.setMap(mapa.map);
},
```

Se crea la función que se ejecutará al pulsar el botón de localizar usuario.

```
getCoords : function()
{
```

Se crea el objeto *geodecoder*

```
 var geocoder = new google.maps.Geocoder();

 address = document.getElementById('search').value;
 if(address!='')
 {
```

Se llama a la función *geodecode* y se le pasa la dirección que hemos introducido en desde la base de datos, es decir la dirección registrada por el usuario al registrarse que fue transformada en coordenadas.

```
 geocoder.geocode({ 'address': address}, function(results, status)
 {
  if (status == 'OK')
  {
```

Se muestran las coordenadas obtenidas en el punto con el identificado de coordenadas

```
  document.getElementById("lat").value= results[0].geometry.location.lat();
  document.getElementById("lng").value= results[0].geometry.location.lng();
```

Se posiciona el marcador en las coordenadas obtenidas

```
  mapa.marker.setPosition(results[0].geometry.location);
```

Y se finalmente se centra el mapa en las coordenadas obtenidas

```
  mapa.map.setCenter(mapa.marker.getPosition());
  agendaForm.showMapaEventForm();
  }
 });
```

3.5.4. Sistema de calificación

Se implementa un sistema de calificación de usuario ofertantes y usuarios no ofertantes en donde se podrá calificar el servicio recibido y el al cliente que lo solicito, este sistema de calificaciones se realizará en base a una escala de 5 puntos donde uno será la calificación más baja y cinco la calificación más alta.

Este sistema tiene la función además de servir de referencia de la calidad del servicio para los usuarios también se utiliza como un sistema con el cual se genera la lista de usuarios destacados que serán expuestos directamente en la sección preparada para esta función en la plataforma, esta lista se genera a partir de un promedio generado por los puntos obtenidos de las calificaciones de cada servicio siendo de esta manera seleccionados automáticamente los servicios con una puntuación superior a cuatro punto siete de la escala antes mencionada de uno a cinco.

El sistema también aporta con un sistema similar para definir usuarios no deseados en la plataforma exponiendo a los servicios con una calificación promedio de menos de uno punto cinco permitiendo al administrador tomar acciones con dichos ofertantes, comunicándose con ellos y posteriormente eliminándolos de la plataforma manteniendo el registro de sus datos de manera independiente para controlar un nuevo registro en la plataforma con los mismos datos.

3.5.5. Sistemas de Notificación

Se implementa un sistema de notificación por correo electrónico y por mensaje de texto el cual está encargada de enviar las notificaciones a los usuarios cuando dentro de la plataforma se les a realizado una solicitud de servicio, así como este encargado de notificar al usuario solicitante que se atendió la solicitud realizada por el mismo, para cumplir con este propósito se trabaja con las librerías SMS de Twilio una plataforma encargada de brindar el servicio de mensajería de forma internacional, para optar por este servicio se tuvo que obtener previamente una *KEY* individual para la plataforma y posteriormente se desarrolló el código necesario para el cumplimiento de esta función.

3.5.6. Funciones de la plataforma

Para el correcto funcionamiento de la plataforma se crearon varias funciones que ayudan en las acciones que se pueden realizar dentro de la plataforma, estas funciones creadas en lenguaje PHP en conjunto con los diferentes módulos de la plataforma se encargan de la validación de datos, control de acceso, restricciones de usuarios y acciones disponibles para los diferentes tipos de usuarios de la plataforma, entre las funciones creadas se pùeden poner como ejemplo las siguientes:

Validación de datos

Se encargará de validar la información que se va a ingresar a la plataforma, también valida que toda la información haya sido completada adecuadamente y cumpliendo las condiciones propuestas para el ingreso de la información; valida también que no se generen usuarios con los mismos datos clave.

A continuación, se exponen algunos ejemplos de este tipo de funciones creadas para la plataforma.

- Validación de datos de registro

```
function isNull($nombre, $apellido, $pass, $pass_con, $email){
    if(strlen(trim($nombre)) < 1 || strlen(trim($apellido)) < 1 || strlen(trim($pass)) < 1 || strlen(trim($pass_con)) < 1 ||
strlen(trim($email)) < 1)  {
            return true;
    } else {
            return false;
    }
}
```

- Validación de formato e-mail

```
function isEmail($email) {
    if (filter_var($email, FILTER_VALIDATE_EMAIL)){
            return true;
    } else {
            return false;
    }
}
```

- Validación de e-mail duplicado

```
function emailExiste($email)        {
    global $mysqli;
    $stmt = $mysqli->prepare("SELECT id FROM tbl_usuarios WHERE correo = ? LIMIT 1");
```

```
        $stmt->bind_param("s", $email);
        $stmt->execute();
        $stmt->store_result();
        $num = $stmt->num_rows;
        $stmt->close();
if ($num > 0){
                return true;
        } else {
                return false;
        }
}
```

Control de acceso

Se encarga de mantener a los usuarios en sus respectivos entornos y controlar que solo los usuarios registrados adecuadamente puedan acceder a la plataforma, así también, de controlar el cumplimiento total del proceso de registro.

A continuación, se exponen algunos ejemplos de este tipo de funciones creadas para la plataforma.

- Controla que solo los usuarios que completaron la verificación de su cuenta tengan acceso a la plataforma.

```
function isActivo($email)
        {
        global $mysqli;
        $stmt = $mysqli->prepare("SELECT activacion FROM tbl_usuarios WHERE correo = ? LIMIT 1");
        $stmt->bind_param('s', $email);
        $stmt->execute();
        $stmt->bind_result($activacion);
        $stmt->fetch();
if ($activacion == 1)
        {
                return true;
        }else{
                return false;
        }
}
```

- Controla que solo los usuarios registrados y verificados tengan acceso a la plataforma.

```
function login($email, $password)
        {
        global $mysqli;
```

```
        $stmt = $mysqli->prepare("SELECT id, id_tipo, password FROM tbl_usuarios WHERE correo = ? LIMIT 1");
        $stmt->bind_param("s", $email);
        $stmt->execute();
        $stmt->store_result();
        $rows = $stmt->num_rows;
    if($rows > 0) {
    if(isActivo($email)){
        $stmt->bind_result($id, $id_tipo, $passwd);
        $stmt->fetch();
        $validaPassw = password_verify($password, $passwd);
    if($validaPassw){
        lastSession($id);
        $_SESSION['id_usuario'] = $id;
    $_SESSION['tipo_usuario'] = $id_tipo;
    header("location: Bienvenido.php");
} else {
        $errors = "La contrase&ntilde;a es incorrecta";
    }} else {
        $errors = 'El usuario no esta activo';
    }} else {
        $errors = "El correo electr&oacute;nico no existe";
    }
    return $errors;
}
```

Acciones de usuarios

Este código se activa cuando los usuarios realizar acciones dentro de sus respectivos entornos.

- Permite al usuario cambiar su contraseña.

```
function cambiaPassword($password, $user_id, $token){
    global $mysqli;
    $stmt = $mysqli->prepare("UPDATE tbl_usuarios SET password = ?, token_password = '', password_request = 0
WHERE id = ? AND token_password = ?");
    $stmt->bind_param('sis', $password, $user_id, $token);
    if($stmt->execute()){
            return true;
    } else {
            return false;
    }}
```

Restricciones

Este código restringe acciones que no hay sido aprobadas o verificadas por los procesos de la plataforma.

- Restringe el acceso a quienes no han completado los datos requeridos para el registro total en la plataforma.

```
function isNullOfertante($idof, $lista_carr, $lista_esp, $radio, $telefono1, $telefono2, $cedula, $lista_prov, $lista_ciu, $direccion, $postal, $referencia, $lat, $lng, $token){
if(strlen(trim($idof)) < 1 || strlen(trim($lista_carr)) < 1 || strlen(trim($lista_esp)) < 1 || strlen(trim($radio)) < 1 || strlen(trim($telefono1)) < 1 || strlen(trim($telefono2)) < 1 || strlen(trim($cedula)) < 1 || strlen(trim($lista_prov)) < 1 || strlen(trim($lista_ciu)) < 1 || strlen(trim($direccion)) < 1 || strlen(trim($postal)) < 1 || strlen(trim($referencia)) < 1 || strlen(trim($lat)) < 1 || strlen(trim($lng)) < 1 || strlen(trim($token)) < 1)
    {
    return true;
    } else {
    return false;
    }
}
```

Las funciones ejemplificadas en el punto anterior trabajan en conjunto con el código implementado en la plataforma para lograr una experiencia adecuada de los usuarios

3.5.7. Login / inicio de sesión

Esta sección se encarga de verificar si un usuario se encuentra registrado en la plataforma al mismo tiempo que determina si el usuario ya inicio sesión desde el navegador específico del dispositivo o equipo que está usando. Si el usuario ya ha iniciado sesión se le redirecciona automáticamente a la página de bienvenida, desde la cual podrán acceder a todas las funciones de la plataforma.

- Código

```
if(!empty($_POST))
    {
        $email = $mysqli->real_escape_string($_POST['email']);
        $password = $mysqli->real_escape_string($_POST['password']);
    if(isNullLogin($email, $password)){
        $errors[] = "Debe llenar todos los campos";
        }
        $errors[] = login($email, $password);
    }
```

3.5.8. Registrar Usuario

Esta sección de código se encarga de ingresar los datos de los nuevos usuarios a la plataforma, este variará dependiendo el tipo de usuario que se esté registrando,

todo esto se realiza previo la selección de opciones de registro la cual dará acceso a el formulario correspondiente del cual se obtendrá los datos que requiere el registro de usuarios.

- Código

```
if(!empty($_POST))
{
	$nombre = $mysqli->real_escape_string($_POST['nombre']);
	$apellido = $mysqli->real_escape_string($_POST['apellido']);
	$email = $mysqli->real_escape_string($_POST['email']);
	$password = $mysqli->real_escape_string($_POST['password']);
	$con_password = $mysqli->real_escape_string($_POST['con_password']);
	$captcha = $mysqli->real_escape_string($_POST['g-recaptcha-response']);
	$activo = 0;
	$tipo_usuario = "Busca";
	$secret = '6LfqZG4UAAAAAA4DFzGzmvfFq0COQJiJYBNWtUGC';//Modificar
if(isNull($nombre, $apellido, $email, $password, $con_password)){
		$errors[] = "Debe llenar todos los campos";
	}
if(!isEmail($email)){
		$errors[] = "Dirección de correo inválida";
		}
if(!validaPassword($password, $con_password)){
		$errors[] = "Las contraseñas no coinciden";
		}
if(emailExiste($email)){
		$errors[] = "El correo electronico $email ya existe";
		}
if(count($errors) == 0){
		$registro = registraUsuario($nombre, $apellido, $email, $pass_hash, $activo, $token, $tipo_usuario);
		} else {
			$erros[] = "Error al enviar Email";
		}} else {
			$errors[] = "Error al Registrar";
		$pass_hash = hashPassword($password);
		$token = generateToken();
	}
}
```

3.5.9. Búsqueda

El código expuesto en el siguiente anexo es el código raíz de las búsquedas, el cual es modificado para que pueda realizar las búsquedas directas de cada una de las diferentes carreras destacadas.

El método de búsqueda cuenta con un bloque que impide que los usuarios con la misma especialidad puedan visualizarse entre sí, impidiendo que se genere

información falsa por parte de solicitudes de servicio atendidas y calificadas de forma negativa.

- Código

```
$Buscar = "SELECT * FROM tbl_usuarios INNER JOIN tbl_ofertantes ON tbl_usuarios.id=tbl_ofertantes.idof WHERE (tbl_usuarios.id_tipo != 'Admin' AND tbl_usuarios.id!='$idUsuario') ORDER By tbl_usuarios.nombre ASC";

if (isset($_POST['consulta'])) {
$Buscar = "SELECT * FROM tbl_usuarios INNER JOIN tbl_ofertantes ON tbl_usuarios.id=tbl_ofertantes.idof WHERE (tbl_usuarios.id_tipo != 'Admin' AND tbl_usuarios.id!='$idUsuario') AND (nombre LIKE '%$q%' OR apellido LIKE '%$q%' OR titulo LIKE '$q%' OR especialidad LIKE '%$q%' OR ciudad LIKE '%$q%') ORDER BY tbl_usuarios.id ";
}
      $resultado_B = $mysqli->query($Buscar);
if ($resultado_B->num_rows>0) {
      while ($fila = $resultado_B->fetch_array()) {
         $id = $fila['idof'];
         $nombre = $fila["nombre"];
         $apellido = $fila['apellido'];
         $car = $fila['carrera'];
         $especialidad = $fila['especialidad'];
         $prov = $fila['provincia'];
         $foto = $fila["foto"];

         $carrera = ("SELECT nombre FROM tbl_carreras INNER JOIN tbl_ofertantes ON tbl_carreras.id_c= '$car' ");
         $res_carr = $mysqli->query($carrera);
         $v_carr = $res_carr->fetch_assoc();
         $vc = $v_carr['nombre'];
    $provincia = ("SELECT nombre FROM tbl_provincias INNER JOIN tbl_ofertantes ON tbl_provincias.idP= '$prov' ");
         $res_prov = $mysqli->query($provincia);
         $v_prov = $res_prov->fetch_assoc();
         $vp = $v_prov['nombre'];
   $salida.="
    <form action= 'referencias.php' method='POST' id='contacto' name='contacto'>
                 <div class='col-md-3 text-center'>
    <table>
          <tr><img class='foto' src='http://minfo.com.ec/images/fotos/$id/$id' alt='$nombre'> </tr>
          <tr><div class='perfil_caption'>
              <h2>$nombre <br> $apellido</h2>
              <h5><span>$vc,</span></h5><h4>$especialidad</h4><br>
                   <button type='submit' name='id_user' id='id_user' class='btn btn-default submit-btn-buscar' value=$id>Referencias</button>
          </div>
           </tr></tr></table>
          </div>
    </form>";
    }$salida.="</tbody></table>";
   }else{
    $salida.="NO HAY DATOS !!!";
```

```
    }
echo $salida;
$mysqli->close();
```

El código de búsqueda expuesto se trabajó para que realizara búsquedas dinámicas con el ingreso de texto relacionado a lo que se busca, es decir que no se recargue la página del módulo para mostrar los resultados de la búsqueda realizada, para lo cual fue necesario implementar otra serie de código distinta que se expone a continuación,

- Código

```
$(buscar_datos());
function buscar_datos(consulta){
    $.ajax({
            url: 'buscar.php' ,
            type: 'POST' ,
            dataType: 'html',
            data: {consulta: consulta},
    })
    .done(function(respuesta){
            $("#datos").html(respuesta);
    })
    .fail(function(){
            console.log("error");
    });
}
$(document).on('keyup','#caja_busqueda', function(){
    var valor = $(this).val();
    if (valor !== "") {
            buscar_datos(valor);
    }else{
            buscar_datos();
    }
});
```

3.5.10. Modificar Datos de Usuario

Esta sección de código pertenece a los datos obtenidos del formulario modificar usuario, que permite la actualización o corrección de la información según sea necesario y la inserta instantáneamente en la base de datos pudiendo observarla al instante sin necesidad de actualizar la página.

- Código

```
$sql = "UPDATE tbl_usuarios INNER JOIN tbl_ofertantes
      ON tbl_usuarios.id = tbl_ofertantes.idof SET
                  tbl_usuarios.nombre = '$nombre', tbl_usuarios.apellido= '$apellido', tbl_ofertantes.carrera= '$lista_carr',
                  tbl_ofertantes.especialidad= '$lista_esp', tbl_ofertantes.telefono1= '$telefono1',
                  tbl_ofertantes.telefono2= '$telefono2', tbl_ofertantes.cedula= '$cedula',
                  tbl_ofertantes.provincia= '$lista_prov', tbl_ofertantes.ciudad= '$lista_ciu',
                  tbl_ofertantes.direccion= '$direccion', tbl_ofertantes.referencia= '$referencia',
                  tbl_ofertantes.lat= '$lat', tbl_ofertantes.lng= '$lng'
                  WHERE tbl_usuarios.id = '$idUsuario' ";

      if ($mysqli->query($sql) === TRUE) {
         echo "Record updated successfully";
         header("location: Bienvenido.php");
      } else {
         echo "Error updating record: " . $mysqli->error;
      }
      $mysqli->close();
```

3.5.11. Eliminar Usuario

En el anterior anexo se puede observar el código de ejecución de la función de eliminar usuarios la cual permitirá a los usuarios darse de baja de la plataforma o a su vez eliminar usuarios que no cumplan con las políticas y condiciones de la misma.

- Código

```
if(isset($_POST['delete'])){
      foreach ($_POST['delete'] as $id):
      $borrar  = BorrarUsuario($id);
      endforeach;
      }

function BorrarUsuario($id){
      global $mysqli;
      $stmt = $mysqli->prepare('DELETE FROM tbl_usuarios WHERE id= ? ');
      $stmt->bind_param('i', $id);
if ($stmt->execute()){
      return $mysqli->insert_id;
} else {
      return 0;
}}
```

3.5.12. Funciones dinámicas

En los anexos siguientes se expone el código las funciones dinámicas que permiten al usuario tener una mejor experiencia al manejar la plataforma, así como reduce las interacciones innecesarias de los usuarios con los módulos de la plataforma.

- Código Ajax Select Dinámicos

```
$(document).ready(function(){
$.ajax({
type: 'POST',
      url: '../includes/cargar_provincias.php'
 })
      .done(function(cargar_provincias){
      $('#lista_prov').html(cargar_provincias)
 })
.fail(function(){
      alert('Hubo un error al cargar las listas_provincias')
 })

 $('#lista_prov').on('change', function(){
 var id = $('#lista_prov').val()
 $.ajax({
      type: 'POST',
      url: '../includes/cargar_ciudades.php',
      data: {'id': id}
})
      .done(function(cargar_ciudades){
      $('#lista_ciu').html(cargar_ciudades)
   })
   .fail(function(){
     alert('Hubo un error al cargar los lista_ciu')    }) })
})
```

La función mostrada en el código anterior es la encargada de la carga instantánea de la información de las tablas de provincia y ciudades así con de carreras y especialidad en una de sus modificaciones; esta función trabaja directamente con las siguientes porciones de código.

- Código

```
require_once '../Connections/conexion.php';
function cargar_provincias()
{
  $mysqli = getConn();
  $query = 'SELECT * FROM `tbl_provincias` ORDER BY nombre ASC';
```

```
  $result = $mysqli->query($query);
  $listas_p = '<option value="0">Provincia</option>';
  while($row = $result->fetch_array(MYSQLI_ASSOC))
  {
    $listas_p .= "<option value='$row[idP]'>$row[nombre]</option>";
  }
  return $listas_p;
}
echo cargar_provincias();
```

Este fragmento de código de encarga de realizar la consulta de la información almacenada en la tabla de provincias y proporcionar esta información al archivo función.js que se encargara de agregar esta información de forma dinámica a el formulario necesario.

- Código

```
require_once '../Connections/conexion.php';
function cargar_ciudades()
{
  $mysqli = getConn();
  $id = $_POST['id'];
  $query = "SELECT idc, idP, ciudad FROM `tbl_ciudades` WHERE idP = $id ORDER BY ciudad ASC";
  $result = $mysqli->query($query);
  $listas_c = '<option value="0">Elige una ciudad</option>';
  while($row = $result->fetch_array(MYSQLI_ASSOC))
  {
    $listas_c .= "<option value='$row[ciudad]'>$row[ciudad]</option>";
  }
  return $listas_c;
}
echo cargar_ciudades();
```

Una vez ejecutada la acción anterior la información obtenida por esta consulta es insertada en el formulario de acuerdo con los datos previamente seleccionados en el formulario en cuestión.

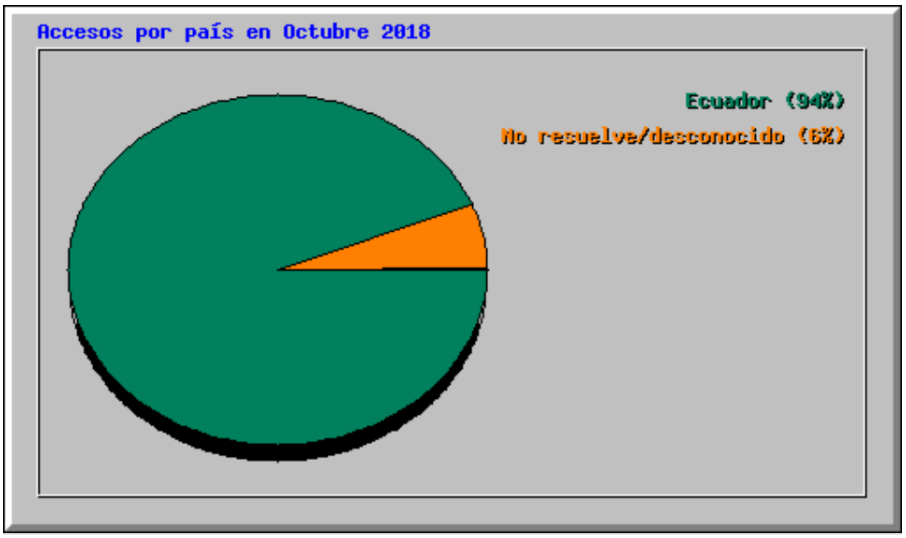
3.5.13. Análisis y Validación de los Resultados

Se expone en este literal argumentos, exámenes e ilustraciones que validan el funcionamiento de la plataforma *online* mostrando datos y obtenidos con software.

A continuación, se exponen tablas y gráficos generados por *webalizer* 2.23 [1] sobre las estadísticas de usos de la plataforma en donde se observa y se contrasta el mes con mayor éxito de la plataforma y el mes con el menor éxito.

Mes de mayor éxito

Ilustración 60, Grafica pastel - Accesos por país - mes más exitoso



Fuente: Elaboración Propia

Ilustración 61, Acceso de los 4 primeros países - mes más exitoso

Los 4 primeros de un total de 4 países							
#	Accesos		Archivos		KBytes		País
1	14548	94.14%	12632	94.28%	84280	87.70%	Ecuador
2	859	5.56%	735	5.49%	11614	12.09%	No resuelve/desconocido
3	45	0.29%	31	0.23%	203	0.21%	Comercial (com)
4	2	0.01%	0	0.00%	0	0.00%	European Union

Fuente: Elaboración Propia

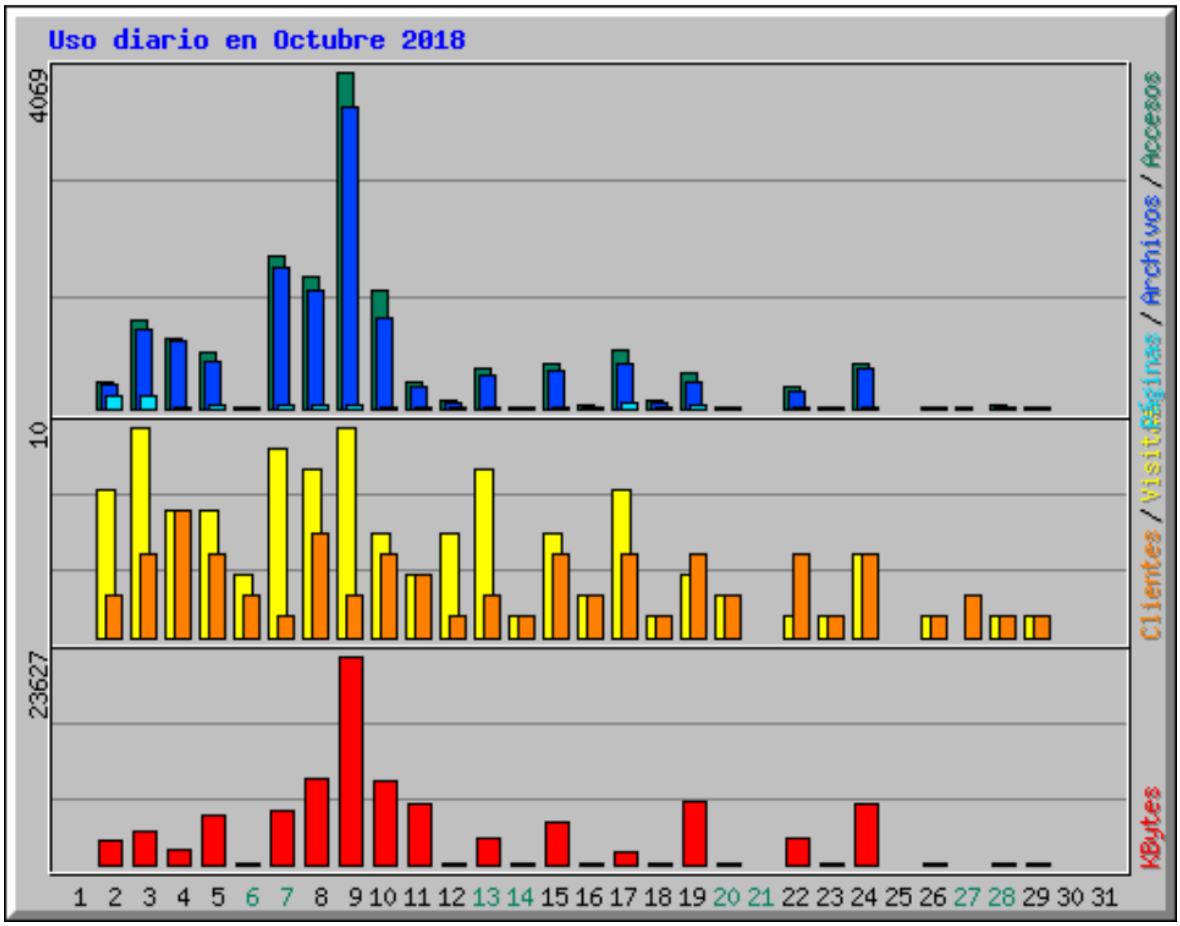
[1] El *Web*alizer es un programa rápido y gratuito de análisis de archivos de registro de servidores *web*. Produce informes de uso altamente detallados y fácilmente configurables en formato HTML, para verlos con un navegador *web* estándar.

Ilustración 62, Estadística mensual - mes más exitoso

Estadísticas mensuales de Octubre 2018		
Total Accesos		**15454**
Total Archivos		**13398**
Total Páginas		**684**
Total Visitas		**108**
Total KBytes		**96097**
Total URLs		**117**
Total Enlaces origen		**81**
Total Navegadores		**17**
	Media	**Max**
Accesos por Hora	**22**	**468**
Accesos por Día	**551**	**4069**
Archivos por Día	**478**	**3647**
Páginas por Día	**24**	**156**
Clientes por Día	**1**	**6**
Visitas por Día	**3**	**10**
KBytes por Día	**3432**	**23627**
Accesos por código de respuesta		
200 - OK	86.70%	**13398**
301 - Movido permanentemente	0.12%	**18**
302 - Encontrado	3.11%	**480**
304 - No modificado	3.09%	**477**
400 - Petición errónea	0.01%	**2**

Fuente: Elaboración Propia

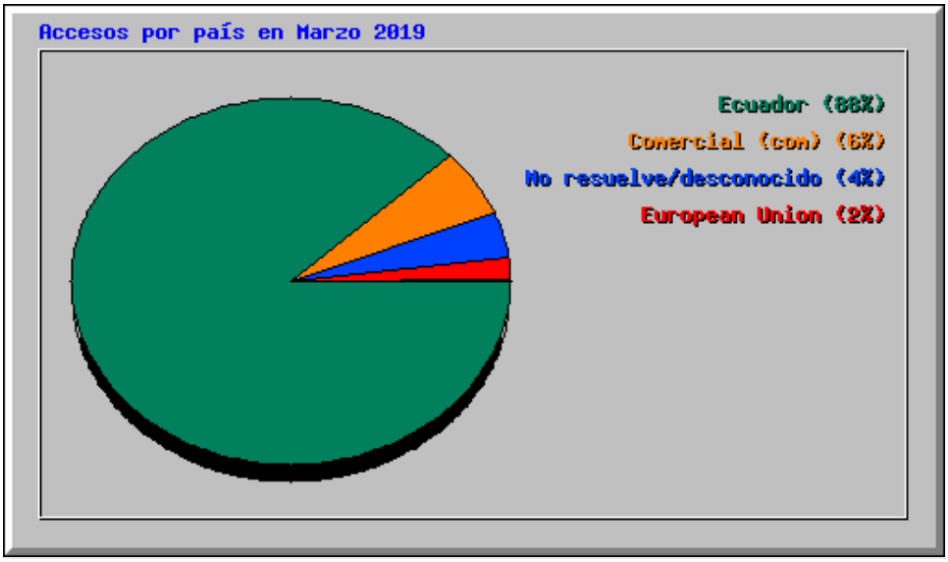
Ilustración 63, Grafico en barras uso diario - mes más exitoso



Fuente: Elaboración Propia

Mes de menor éxito

Ilustración 64, Grafica pastel - Accesos por país - mes menos exitoso



Fuente: Elaboración Propia

Ilustración 65, Acceso de los 4 primeros países - mes menos exitoso

Los 5 primeros de un total de 5 países							
#	Accesos		Archivos		KBytes		País
1	252	87.50%	151	100.00%	4586	96.98%	Ecuador
2	17	5.90%	0	0.00%	0	0.00%	Comercial (com)
3	11	3.82%	0	0.00%	142	3.01%	No resuelve/desconocido
4	7	2.43%	0	0.00%	0	0.01%	European Union

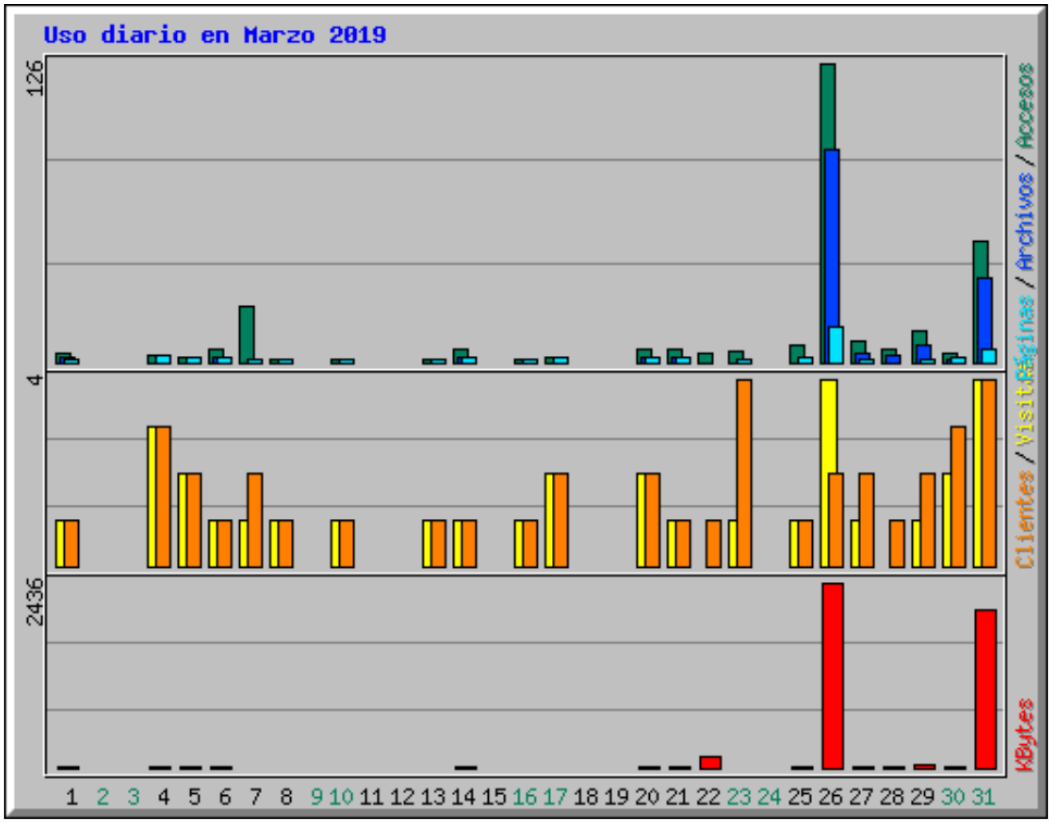
Fuente: Elaboración Propia

Ilustración 66, Estadística mensual - mes menos exitoso

Estadísticas mensuales de Marzo 2019		
Total Accesos		288
Total Archivos		151
Total Páginas		49
Total Visitas		32
Total KBytes		4728
Total Clientes		27
Total URLs		50
Total Enlaces origen		18
Total Navegadores		11
	Media	Max
Accesos por Hora	0	64
Accesos por Día	9	126
Archivos por Día	4	90
Páginas por Día	1	15
Clientes por Día	0	4
Visitas por Día	1	4
KBytes por Día	153	2436
Accesos por código de respuesta		
200 - OK	52.43%	151
301 - Movido permanentemente	0.69%	2
302 - Encontrado	1.04%	3
304 - No modificado	11.81%	34
401 - No autorizado	4.86%	14
404 - No se encuentra	29.17%	84

Fuente: Elaboración Propia

Ilustración 67, Grafico en barras uso diario - mes menos exitoso



Fuente: Elaboración Propia

3.5.14. Certificado *Secure Socket Layer*

Este certificado valida la seguridad que brinda la plataforma *online* al usuario, contar con él, avala que la plataforma *online* es auténtica y confiable para registrar tus datos personales.

Es un protocolo de seguridad certifica que sus datos ingresado es la plataforma se trasladan de manera íntegra y segura, lo que se puede traducir como, la transmisión de los datos entre el usuario de la plataforma y el servidor es totalmente cifrada o encriptada, al mencionar esto se hace referencia a que se emplean algoritmos matemáticos y un sistema de claves que solo pueden ser leído por el usuario en la plataforma y el servidor. Al tener este certificado los datos de los usuarios están encriptados, y podemos aseverar que nadie puede leer su contenido fuera de la plataforma.

Este certificado implementa y valida un modelo con preferencia en la seguridad de la plataforma *online*, trabaja con contraseñas que permites la protección y la integridad de los datos de los usuarios al momento de registrase en la plataforma. Las plataformas *web* y los servidores que trabajan con este certificado mantienen un canal con un cifrado único para que los usuarios mantengan sus datos y sus acciones en la plataforma de manera privada, se mantienen de esta forma en la plataforma una clave pública y una clave privada: La clave pública es para cifrar la información, la clave privada para descifrarla.

Para seleccionar el certificado se tomó en cuenta el alcance deseado en la plataforma *online*, la audiencia que accederá y por la sesión que el usuario realizará. En la actualidad la mayoría de las aplicaciones *web* y servidores soportan un certificado SSL es por eso que le recomendamos analizar a profundidad la finalidad de su sitio *web* y haga una excelente decisión en cuanto a certificado SSL se refiere.

Inicio

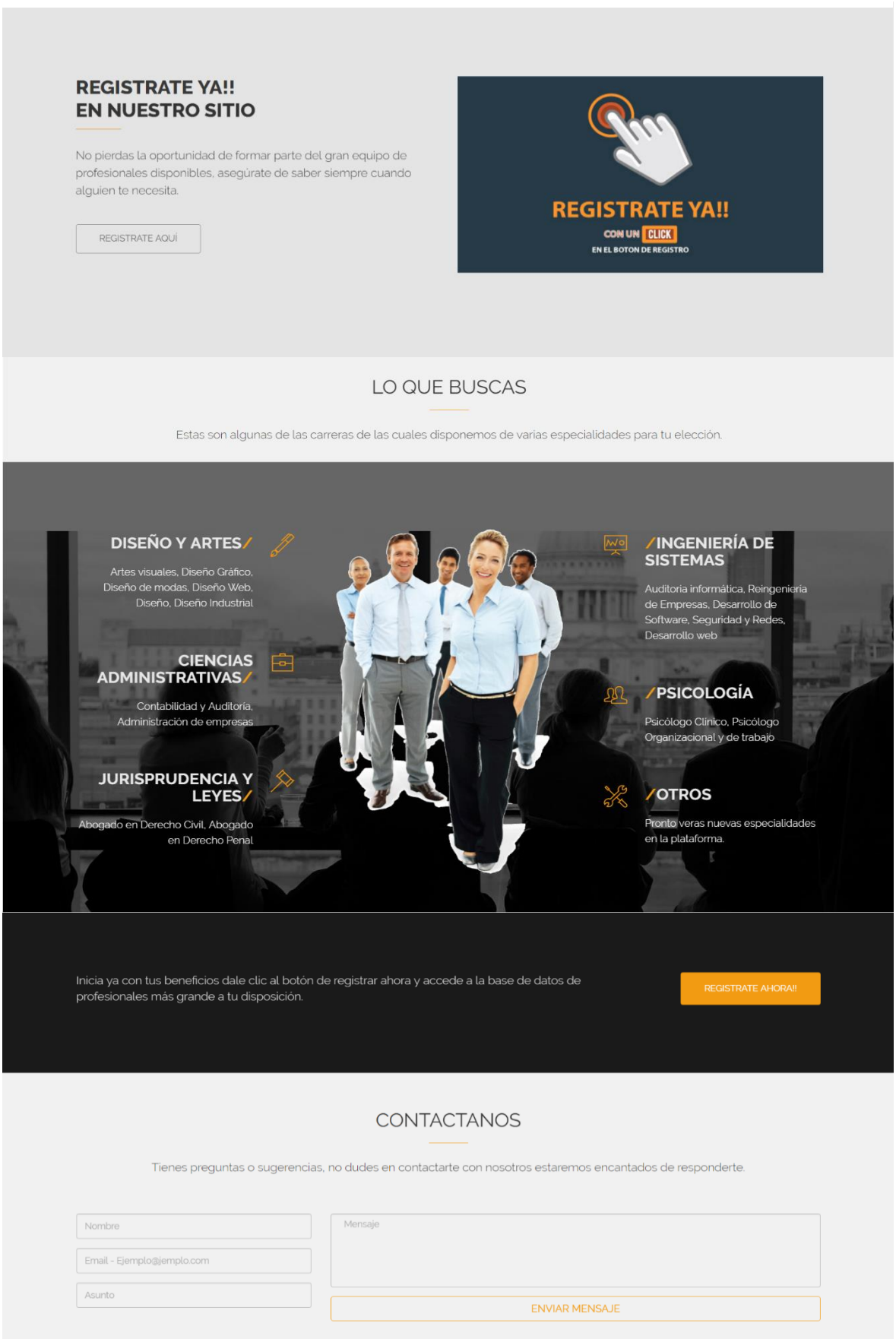
A continuación, se ilustra la página de inicio de la aplicación en la cual se puede encontrar información sobre el funcionamiento de la plataforma y sobre el trabajo que se ha realizado como proyecto

Ilustración 68, Modulo Bienvenidos - Parte 1 de 2



Fuente: Elaboración Propia

Ilustración 69, Modulo Bienvenidos - Parte 2 de 2



Fuente: Elaboración Propia

Tipos de registro

Módulo en que el usuario selecciona el tipo de usuario con el que se desea registrar.

Ilustración 70, Pagina Tipo de Registro



Fuente: Elaboración Propia

Módulo: Registro

Módulo de registro de usuarios

Ilustración 71, Modulo de Registro

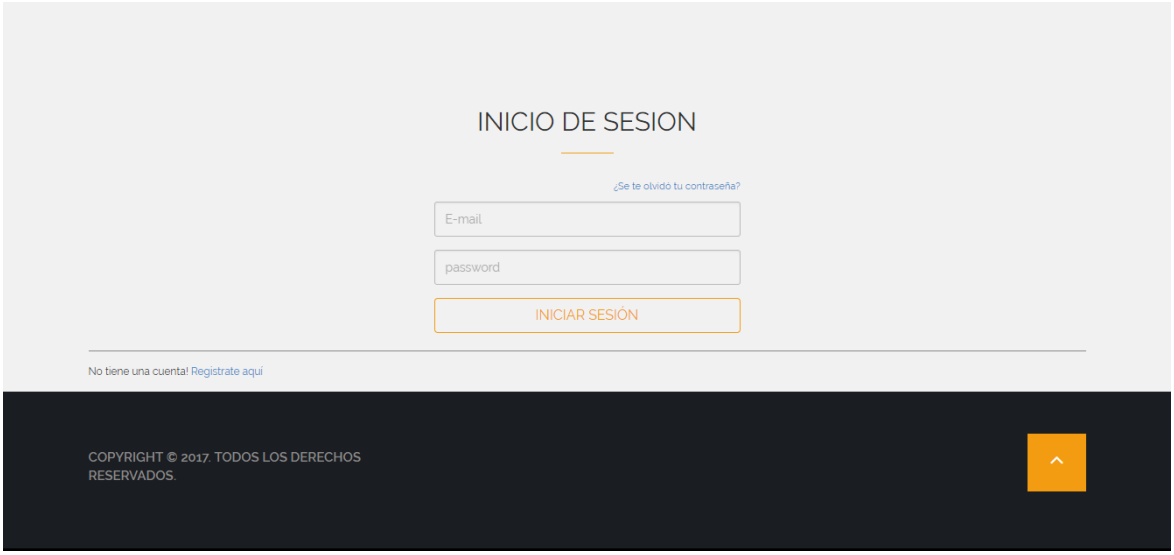


Fuente: Elaboración Propia

Módulo: Logeo / Inicio de sesión

Módulo de logeo de usuarios registrados en la plataforma.

Ilustración 72, Modulo de logeo

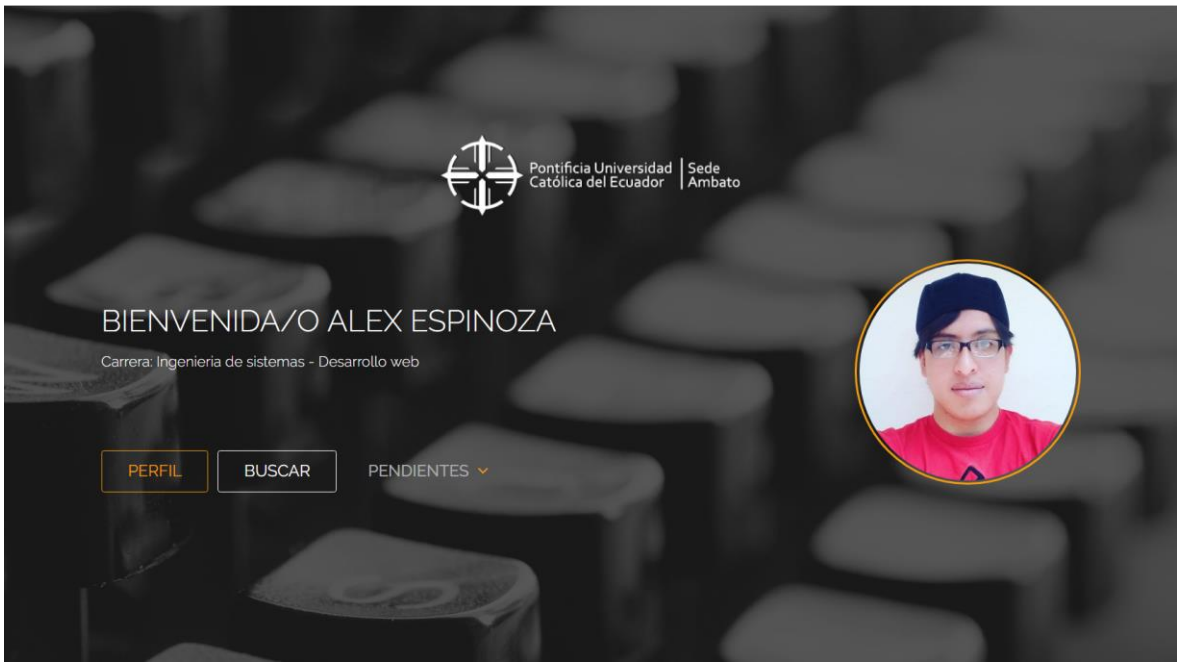


Fuente: Elaboración Propia

Módulo: Bienvenido

Vista inicial de la página de acceso para usuarios registrados dividida varias secciones explicadas a continuación.

Ilustración 73, Pagina Bienvenido Cabecera

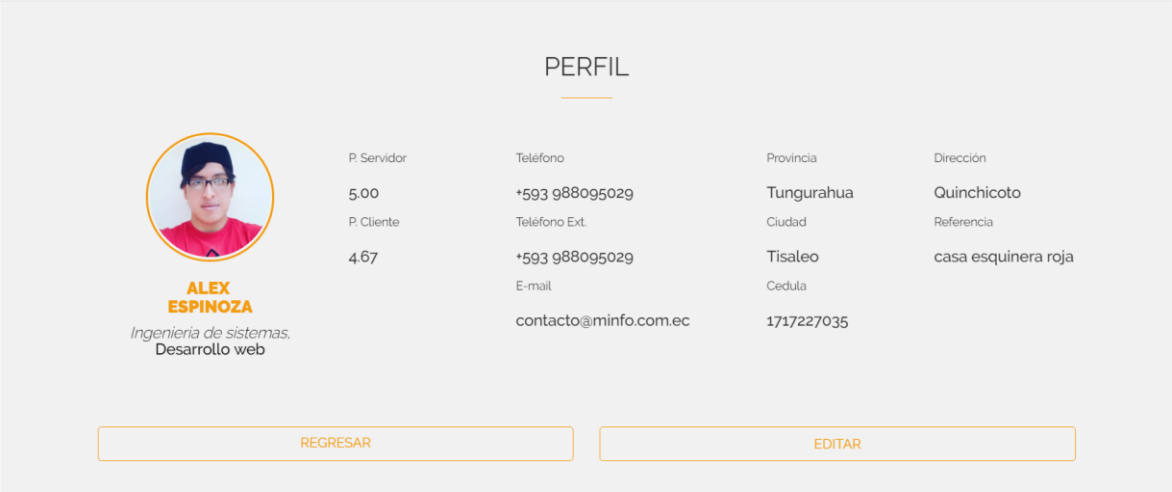


Fuente: Elaboración Propia

Módulo: Perfil

Contiene la información de los usuarios proporcionada por ellos y generada en la plataforma.

Ilustración 74, Modulo de Perfil

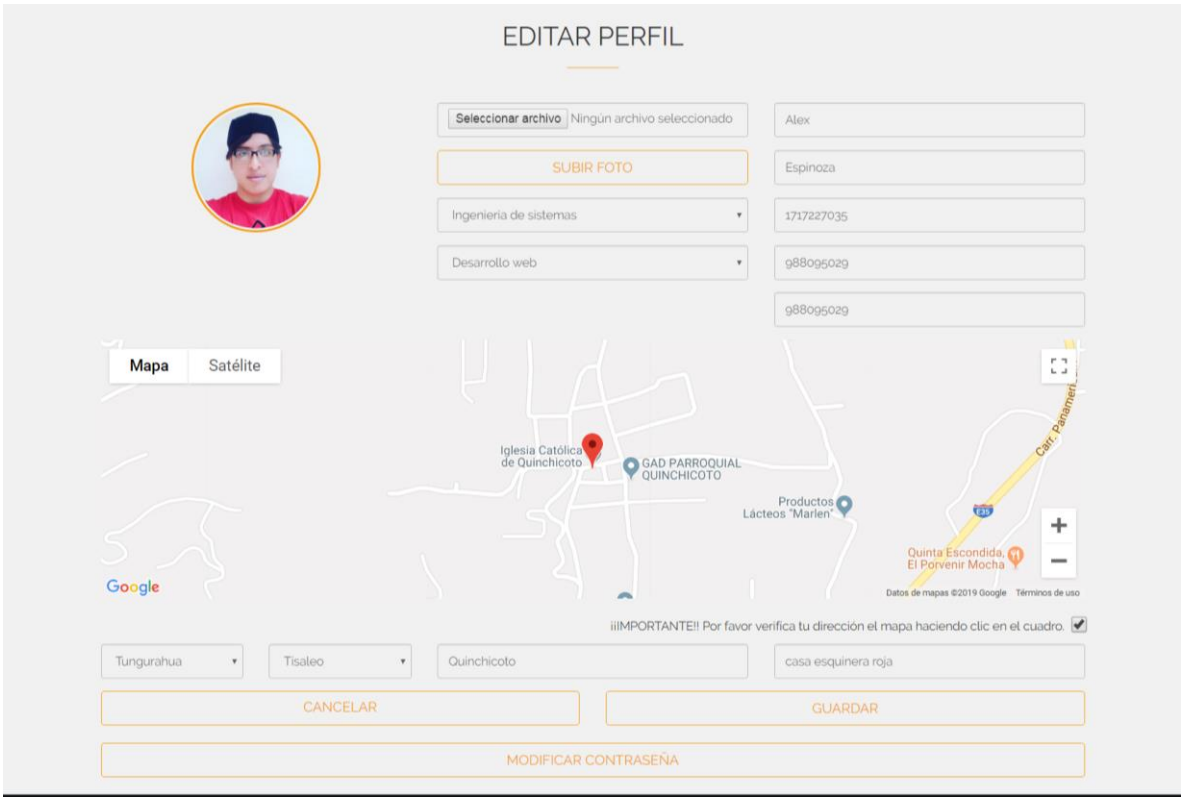


Fuente: Elaboración Propia

Módulo: Edición de Perfil

El módulo para editar los datos del perfil del usuario comparte las mismas funciones tanto para usuarios ofertantes como para usuarios no ofertantes con la diferencia que el usuario no ofertante tendrá menos campos para llenar o modificar.

Ilustración 75, Modulo Edición de Perfil

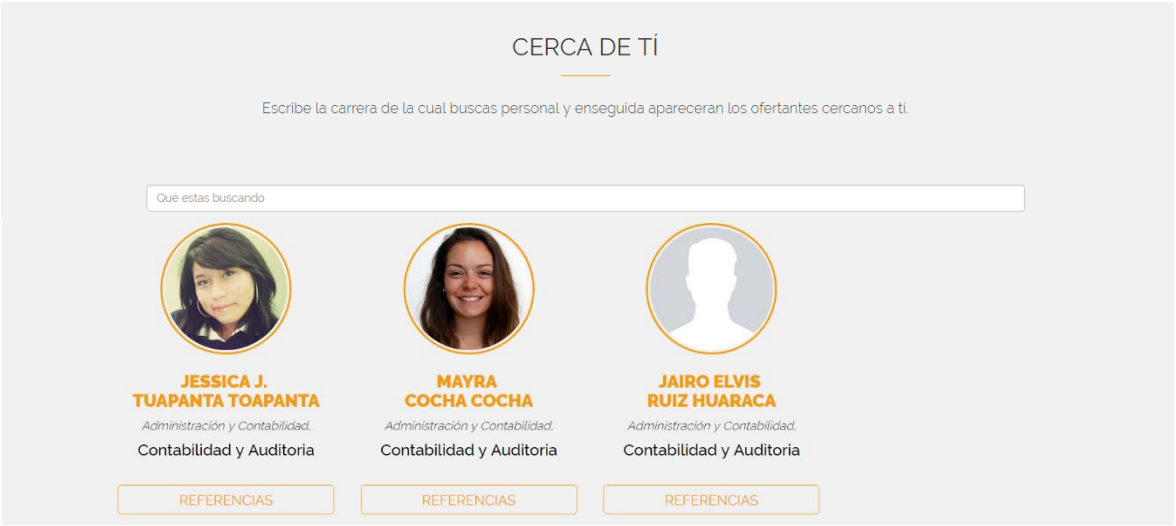


Fuente: Elaboración Propia

Sección: Búsqueda cerca de ti

Esta sección expone los ofertantes registrados en la plataforma más cercanos al usuario que realiza la búsqueda.

Ilustración 76, Modulo Búsqueda Cerca de ti

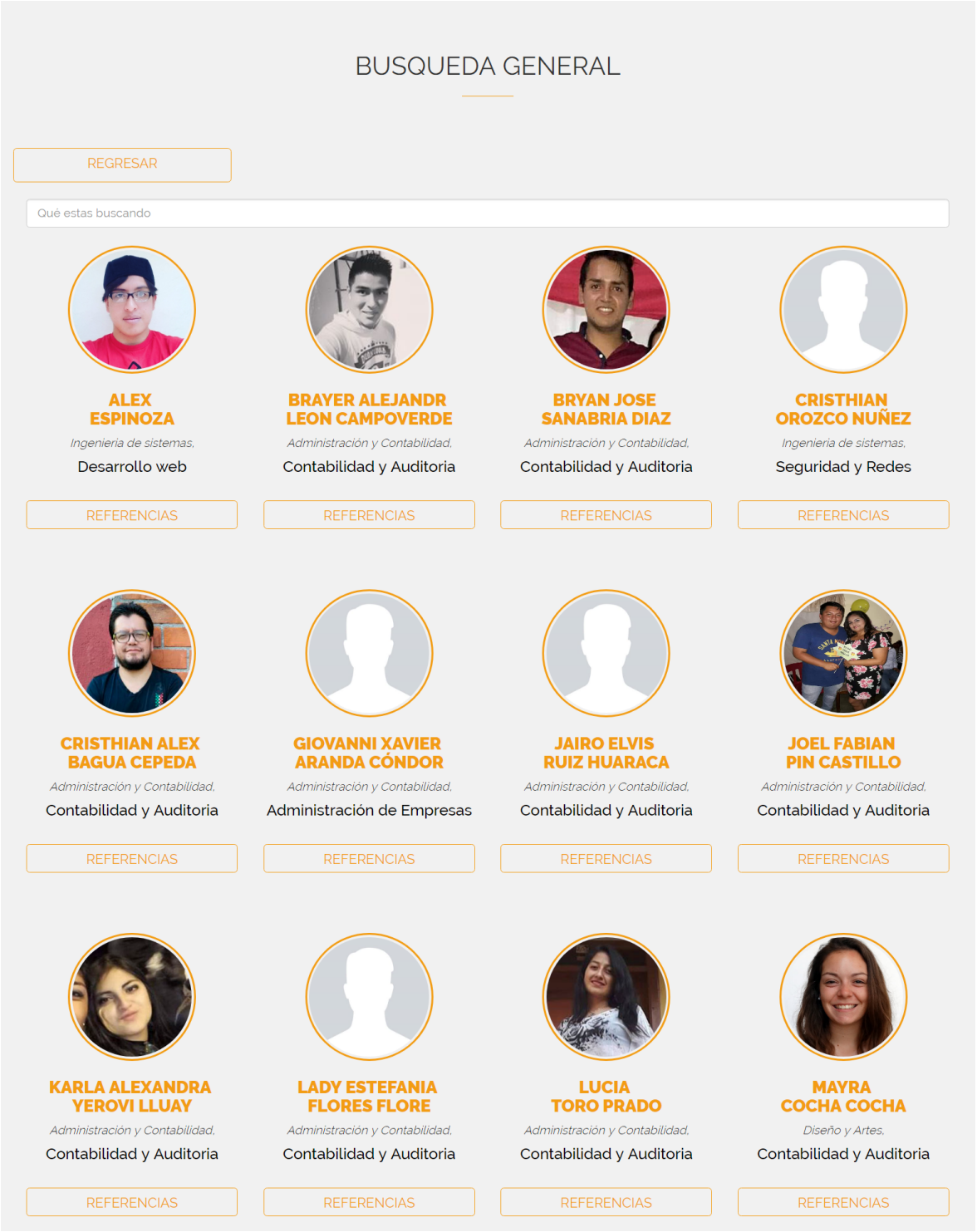


Fuente: Elaboración Propia

Módulo: Búsqueda general

Este módulo expone a todos los ofertantes registrados en la plataforma.

Ilustración 77, Modulo Búsqueda General,



Fuente: Elaboración Propia

Modulo: Referencias

Permite observar las calificaciones y comentarios que obtiene el usuario al concretar una petición de servicio.

Ilustración 78, Modulo Referencia de Ofertante

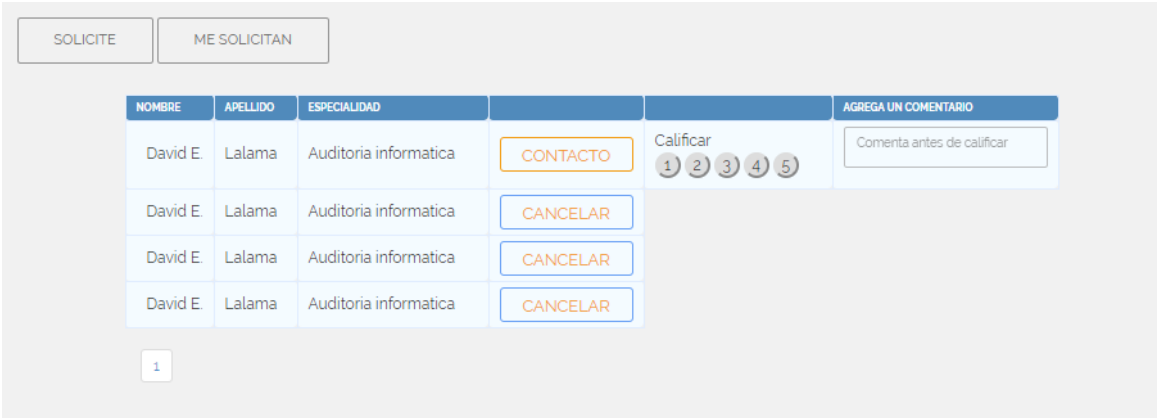


Fuente: Elaboración Propia

Sección: Pendientes/ Solicite

Muestra las peticiones hechas por una cuenta de un usuario ofertante o no ofertante interesado en contratar el servicio que el usuario solicitado proporciona.

Ilustración 79, Sección Solicite Pendientes



Fuente: Elaboración Propia

Sección: Pendientes/ Me solicitan

Muestra las peticiones hechas a una cuenta de ofertante por usuarios interesados en contratar el servicio que el usuario proporciona.

Ilustración 80, Sección Me Solicitan Pendientes



Fuente: Elaboración Propia

Sección: Pendientes/ Solicite o Me solicitan/ Botón Contacto

Permite visualizar la información de los usuarios que han confirmado las solicitudes de servicio.

Ilustración 81, Sección Pendientes Contacto



Fuente: Elaboración Propia

Sección: Destacados

Resalta los usuarios con mejores calificaciones de la plataforma.

Ilustración 82, Sección Destacados



Fuente: Elaboración Propia

Sección: Footer y Eliminar cuenta

Todos los usuarios cuentan con la opción de dar de baja sus respectivas cuentas en la sección inferior de la página de bienvenida.

Ilustración 83, Sección Footer Eliminar cuenta permanentemente

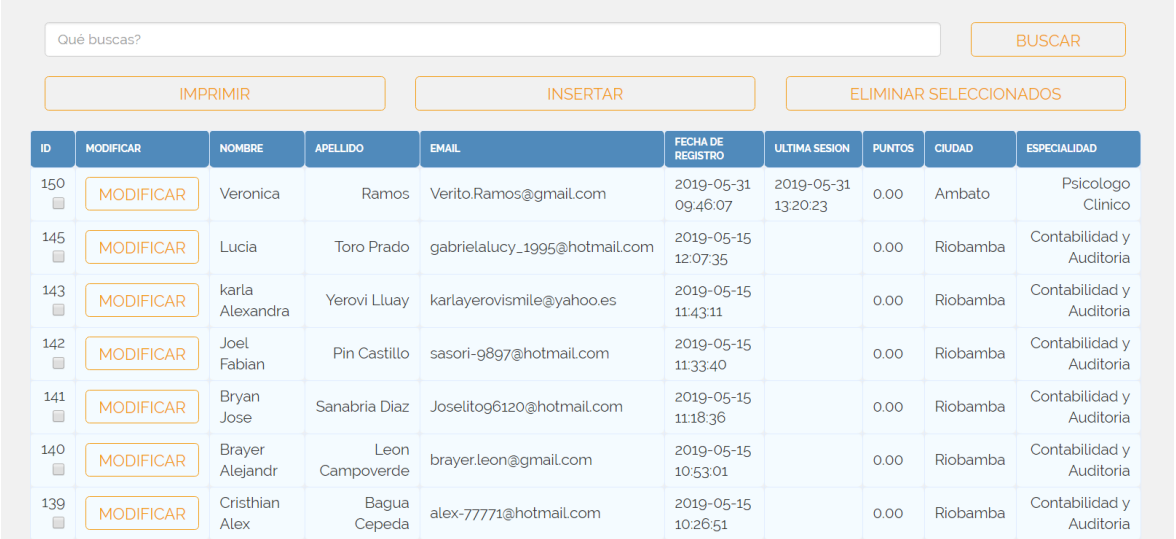


Fuente: Elaboración Propia

Módulo: Administración de Usuarios

Expone todos los usuarios de la plataforma permitiendo gestionar las acciones que el administrador requiera sobre cada uno de ellos como modificar o eliminar.

Ilustración 84, Modulo Administración de Usuarios



ID	MODIFICAR	NOMBRE	APELLIDO	EMAIL	FECHA DE REGISTRO	ULTIMA SESION	PUNTOS	CIUDAD	ESPECIALIDAD
150	MODIFICAR	Veronica	Ramos	Verito.Ramos@gmail.com	2019-05-31 09:46:07	2019-05-31 13:20:23	0.00	Ambato	Psicologo Clinico
145	MODIFICAR	Lucia	Toro Prado	gabrielalucy_1995@hotmail.com	2019-05-15 12:07:35		0.00	Riobamba	Contabilidad y Auditoria
143	MODIFICAR	karla Alexandra	Yerovi Lluay	karlayerovismile@yahoo.es	2019-05-15 11:43:11		0.00	Riobamba	Contabilidad y Auditoria
142	MODIFICAR	Joel Fabian	Pin Castillo	sasori-9897@hotmail.com	2019-05-15 11:33:40		0.00	Riobamba	Contabilidad y Auditoria
141	MODIFICAR	Bryan Jose	Sanabria Diaz	Joselito96120@hotmail.com	2019-05-15 11:18:36		0.00	Riobamba	Contabilidad y Auditoria
140	MODIFICAR	Brayer Alejandr	Leon Campoverde	brayer.leon@gmail.com	2019-05-15 10:53:01		0.00	Riobamba	Contabilidad y Auditoria
139	MODIFICAR	Cristhian Alex	Bagua Cepeda	alex-77771@hotmail.com	2019-05-15 10:26:51		0.00	Riobamba	Contabilidad y Auditoria

Fuente: Elaboración Propia

Módulo: Modificar Datos de usuarios por Administración de Usuarios

El usuario administrador puede modificar los datos de cualquier usuario en caso de ser necesario accediendo a este módulo.

Ilustración 85, Modulo modificar cuenta por Administración de Usuarios



Fuente: Elaboración Propia

Módulo: Insertar nuevo usuario ofertante

El usuario administrador puede insertar en la base de datos un nuevo usuario

Ilustración 86, Modulo insertar nuevo usuario por Administración de Usuarios



Fuente: Elaboración Propia

3.5.15. Problemas encontrados en la implementación

Durante el desarrollo de la plataforma se encontró algunos problemas como la compatibilidad de las versiones de JQuery y JavaScript pues al inicio del desarrollo de la plataforma se encontraron en vigencia versiones anteriores a las versiones al momento de la implementación de la plataforma, para solucionar estos problemas se debió implementar las últimas versiones de JQuery y JavaScript dentro de los recursos usados por la plataforma, una vez implementados los recursos actuales se procedió a revisar el código de enlace en las diferentes paginas desarrolladas.

Entre otros problemas una vez implementado el proyecto se presentaron inconvenientes económicos externos que impidieron el continuo desarrollo y cumplimiento de la meta del proyecto.

CONCLUSIONES

- El método OOHDM permitió documentar el proceso de desarrollo de la plataforma lo cual facilitó la comprensión de cada una de las etapas del método al mismo tiempo que simplificó la implementación del proyecto.
- El uso de *framework* resulta muy conveniente y ayuda a mantener un ritmo de desarrollo fluido en lo que respecta al diseño de la aplicación.
- El uso de las aplicaciones de *Maps JavaScript API* de la plataforma *Google developers* fueron fundamentales facilitando el trabajo al momento de codificar los servicios de geolocalización en la plataforma.
- El sistema de puntuación o calificación es muy necesario en plataformas con características similares en cuanto a manejo de usuarios respecta pues se hace necesario resaltar las cualidades de los usuarios, sin exceder la cantidad de información requerida a los mismos.

RECOMENDACIONES

- Se recomienda ampliamente el método utilizada (OOHDM) pues fue de gran ayuda para la formación paso a paso de la aplicación, sin embargo, es necesario el contraste con otras metodologías más rápidas.
- Es necesario el contrastar las herramientas de desarrollo responsivo disponibles de acuerdo a la planificación de las aplicaciones, algunas opciones de framework proporcionan funciones mejor desarrolladas para modelado de las aplicaciones *web*.
- Se recomienda profundizar los conocimientos de JavaScript y del manejo de permisos dentro de *Google Developers* para así poder aprovecharlos de forma más fluida en el desarrollo de futuras aplicaciones *Web*.
- En todo sistema de puntuación se hace necesario el entregar un porque de dichos puntajes ya sea como en este caso un cometario justificando la puntuación otorgada o un sistema de evaluación de méritos.

BIBLIOGRAFÍA

[1] G. L. José, Gestión de contenidos *web*. Manual teórico., Madrid: ES: Editorial CEP, 2015.

[2] N. G. Carr, ¿Qué está haciendo Internet con nuestras mentes? Superficiales, W. W. Norton & Company, 2010.

[3] V. T. Aranda, «Historia y evolución del internet,» ACTA, [En línea]. Available: http://www.acta.es/medios/articulos/comunicacion_e_informacion/033021.pdf. [Último acceso: 20 10 2016].

[4] C. Aubry, Cree su primer sitio *Web*: del diseño a la realización, ENI, 2012.

[5] J. R. P. Martínez, La World Wide *Web* e Internet. Factores de cambio y constantes transformaciones, GRIN Verlag, 2018.

[6] María Jesús Marco Galindo, Josep María Marco Simó, Escaneando la informática, Barcelona: UOC, 2010.

[7] P. F. D. C. A. Y. D. D. P. W. P. U. S. D. G. D. USUARIOS, «http://docplayer.es/,» [En línea]. Available: http://docplayer.es/3844760-Analisis-y-diseno-de-una-plataforma-*web*-para-un-sistema-de-gestion-de-usuarios.html. [Último acceso: 26 11 2016].

[8] R. O. Griñán, Redes sociales y servicios *Web* 2.0: Comunicación, participación y aprendizaje, Marpadal Interactive Media S.L., 2017.

[9] J. Nño, Aplicaciones *Web* 2.0 (Aplicaciones *web*), Editex, 2011.

[10] A. L. C. Cabello, Implantación de aplicaciones *web* en entornos internet, intranet y extranet. IFCD0210, IC, 2015.

[11] N. P. Christopher Murphy, HTML y CSS, NA: Fiendsoft, 2009.

[12] «The CSS Saga,» (s.f.). [En línea]. Available: https://www.w3.org/Style/LieBos2e/history/. [Último acceso: 15 10 2016].

[13] «Historia de PHP,» (s.f). [En línea]. Available: http://php.net/manual/es/history.php.php. [Último acceso: 15 10 2016].

[14] E. GODOC, SQL fundamentos del Lenguaje, Barcelona: Eni Ediciones, 2014.

[15] «A Short History of JavaScript,» 24 6 2012. [En línea]. Available: https://www.w3.org/community/*web*ed/wiki/A_Short_History_of_JavaScript. [Último acceso: 15 10 2016].

[16] G. Company, «Desarrolladores de Google,» 2018. [En línea]. Available: https://developers.googleblog.com/. [Último acceso: 10 octubre 2018].

[17] Antonio Tena Suck, Rodolfo Rivas-Torres, Manual de investigación documental: elaboración de tesinas, Universidad Iberoamericana, 2009.

[18] M. Barrie, «Freelancer,» 2000. [En línea]. Available: http://www.Freelancer.com,. [Último acceso: 11 9 2016].

[19] Fabrice Grind, Alec Oxenford, «Olx,» 2006. [En línea]. Available: https://www.olx.com.ec/help/about. [Último acceso: 14 9 2016].

[20] LAMBERT M. SORHONE, MARIAM T. TENNOE, SUSAN F. HENSSONOW, S. Systematic Hypermedia Application Design with OOHDM, Betascript, 2010.

[21] Sonia Jaramillo Valbuena, Leonardo Alonso Hernández Rodríguez, Programación orientada a objetos, Armenia-Quindio, 2010. .

[22] J. Hernández, Análisis y Desarrollo *Web*, Jesús Hernández, 2014.

[23] I. Sommerville, Ingeniería del software, Pearson, 2015.

[24] A. B. G. Mariscal, UF2405 - Modelo de programación *web* y bases de datos, Elearning, S.L, 2015.

[25] Bernd Bruegge, Allen H. Dutoit, «Object-Oriented Software Engineering Using UML, Patterns, and Java: Pearson New International Edition,» NA NA 2013. [En línea]. Available: http://dbmanagement.info/Books/MIX/POO_Software_Engineering_Using_UML_Patterns_and_Java_3rd_Edition.pdf. [Último acceso: 10 15 2016].

[26] C. S. Pineda, «Un M´etodo de Desarrollo de Hipermedia Dirigido,» Julio 2008. [En línea]. Available: http://www.dsic.upv.es/docs/bib-dig/tesis/etd-09092008-094903/TesisCSP.pdf. [Último acceso: 14 enero 2017].

[27] R. G. Schulz, Diseño *web* con CSS, Marcombo, 2008.

[28] M. J. E. Cuaresma, «Universidad de Sevilla,» octubre 2001. [En línea]. Available: https://www.lsi.us.es/docs/informes/EstadoActual.pdf. [Último acceso: 10 enero 2017].

[29] Simon Bennett, Steve McRobb, Ray Farmer, Análisis y diseño orientado a objetos de sistemas usando UML, vol. 21, McGraw-Hill, 2008.

[30] E. Gamma, Patrones de diseño: elementos de software orientado a objetos reutilizable 4ta Edicion, Malaga: Pearson Educación, 2008.

[31] J. R. P. Martínez, La World Wide *Web* e Internet. Factores de cambio y constantes transformaciones, GRIN Verlag, 2018.

[32] Coral Calero Muñoz, Mario G. Piattini Velthuis, María Ángeles Moraga de la Rubia, Calidad del producto y proceso software, Ra-Ma, 2010.

[33] J. J. T. Cordero, LA GUÍA DEFINITIVA DEL DISEÑO *WEB*, 2014.

[34] Tiziana Catarci, Alan Dix, Stephen Kimani, Giuseppe Santucci, User-centered Data Management, N/A: Morgan & Claypool , 2010.

[35] LAMBERT M. SURHONE, MARIAM T. TENNOE, SUSAN F. HENSSONOW, OOHDM, Betascript , 2010.

[36] «stackshare,» 13 marzo 2017. [En línea]. Available: https://stackshare.io/stackups/bootstrap-vs-foundation-vs-materialize#more. [Último acceso: 10 octubre 2018].

[37] G. Company, «Google trends,» 1 julio 2018. [En línea]. Available: https://trends.google.com/trends/explore?date=today%2012-m,today%2012-m,today%2012-m&geo=,,&q=Bootstrap,Foundation,Materialize. [Último acceso: 10 octubre 2018].

[38] G. Company, «Google Maps Platform,» 16 Julio 2018. [En línea]. Available: https://developers.google.com/maps/documentation/?hl=es. [Último acceso: 10 octubre 2018].

[39] C. C. Sánchez, Estrategias de comunicación Multimedia, Editorial UOC, 2014.

[40] L. Pairuna, «CodeDimension,» [En línea]. Available: http://www.codedimension.com.ar/noticias-sobre-tecnologia/noticias/-que-es-y-para-que-sirve-un-sitio-*web*-/1. [Último acceso: 14 10 2016].

[41] R. S. Pressman, «Ingeniería *Web*,» de *Ingeniería del Software. Un enfoque practico*, Mc Graw Hill, 2015.

[42] C. M. Razo, Cómo elaborar y asesorar una investigación de tesis, Pearson Educación, 2010.

[43] C. G. Torres, Análisis y Síntesis de Mecanismos con Aplicaciones, Grupo Editorial Patria, 2015.