

UNIVERSIDAD DE CIENCIAS COMERCIALES

UCC – SEDE MANAGUA



COORDINACIÓN DE INFORMÁTICA Y TELECOMUNICACIONES

PROPUESTA PARA LA IMPLEMENTACIÓN DEL FRAMEWORK DE APLICACIONES WEB DE CÓDIGO ABIERTO “RUBY ON RAILS”, ORIENTADO HACIA LA ENTREGA DE INFORMES EDUCATIVOS, EMPRESARIALES U OTROS. REALIZADO EN EL PERIODO DE ENERO A MAYO 2022 EN MANAGUA.

ELABORADO POR:

Ing. Jorge Palacios Linarte

Lic. Pastora Tenorio

ASESOR:

Lic. Octavio Saldaña

MANAGUA, 12 DE JUNIO DEL 2022

RESUMEN

El presente proyecto se refiere al desarrollo de una aplicación en el entorno de trabajo de desarrollo web Ruby on rails. Que se puede definir cómo uno de los entornos más productivos actualmente alcanzando a generar mayor utilidad a los desarrolladores por tiempo. Con un enfoque de investigación mixto se determino que: Los recursos nativos y los desarrollados por terceros son de gran utilidad para implementar funcionalidades que sean de gran rendimiento, de estos recursos existen tantos como aspectos de cualquier proyecto, estos muestran una gran calidad, flexibilidad y un alto desempeño generalizado. Se verifico que la eficiencia y eficacia que genera los recursos de rails en la experiencia de utilizar una aplicación, tanto en campos básicos como en otros un poco más laboriosos, pero también se pudo conocer la estrecha relación que existe entre el éxito de estas herramientas y el correcto diseño en la lógica del desarrollo. Se corrobora de forma específica la facilidad que ofrece rails para poder generar instrumentos como informes y reportes de forma efectiva.

Palabras claves: Ruby on Rails, desarrollo, web, aplicación

ABSTRACT

This project refers to the development of an application in the Ruby on Rails web development work environment. Which can be defined as one of the most productive environments currently generating greater utility for developers per time. With a mixed research approach, it was determined that: Native resources and those developed by third parties are very useful to implement functionalities that are high performance, there are as many of these resources as there are aspects of any project, they show great quality, flexibility and high overall performance. It was verified that the efficiency and effectiveness that rails resources generate in the experience of using an application, both in basic fields and in others that are a little more laborious, but it was also possible to know the close relationship that exists between the success of these tools and the correct design in the development logic. The ease that rails offers to be able to generate instruments such as reports and reports effectively was specifically confirmed.

Keywords: Ruby on Rails, development, web, application

ÍNDICE DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA	2
1.1 Antecedentes	2
1.2 Objetivos	4
1.3 Descripción del problema y Pregunta de Investigación.....	5
1.4 Justificación	5
1.5 Limitaciones	6
1.6. Variables	7
1.7.- Categorías, temas y patrones emergentes de la investigación.....	7
1.8 Supuestos básicos	7
1.9. Contexto de la investigación	8
CAPÍTULO II: MARCO REFERENCIAL	10
2.1. Estado del arte	10
2.2 Perspectiva teórica asumida	11
CAPÍTULO III: DISEÑO METODOLÓGICO	18
3.1 tipo de investigación.....	18
3.1.1 Tipo de diseño de investigación.....	18
3.1.2 Nivel de estudio	18
3.1.3 Paradigma o enfoque de investigación.....	18
3.2 Área de estudio	19
3.2.1 Ubicación, tamaño, institución	19
3.3 Unidades de análisis	20
3.3.1 Tamaño de población teórica.....	20
3.4 Instrumentos de recolección de datos.....	21
3.4.1 Fuente de información primaria	21

3.4.2 Fuentes de información secundaria (publicaciones, sitios web y bases de datos).....	21
3.5 Confiabilidad y validez de los instrumentos	22
3.6 Procesamiento de datos y análisis de la información.....	23
3.7 Operacionalización de variables	30
CAPÍTULO IV: ANÁLISIS DE RESULTADOS.....	31
CAPÍTULO V: CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN.....	32
CAPÍTULO VI: RECOMENDACIONES	33
REFERENCIAS BIBLIOGRÁFICAS	34
ANEXOS	35

INDICE DE TABLA

Tabla 1. Operacionalización de Variables	30
---	----

ÍNDICE DE FIGURA

Figura 1. Pregunta 1 de encuesta.....	25
Figura 2. Pregunta 2 de encuesta.....	25
Figura 3. Pregunta 3 de encuesta.....	26
Figura 4. Pregunta 4 de encuesta.....	26
Figura 5. Pregunta 5 de encuesta.....	26
Figura 6. Pregunta 6 de encuesta.....	27
Figura 7. Pregunta 7 de encuesta.....	27

ÍNDICE DE ANEXO

Anexo 1. Grafico De Pregunta 1 De Encuesta.	35
Anexo 2. Grafico De Pregunta 2 De Encuesta.	35
Anexo 3. Grafico De Pregunta 3 De Encuesta.	35
Anexo 4. Grafico De Pregunta 4 De Encuesta.	36
Anexo 5. Grafico De Pregunta 5 De Encuesta.	36
Anexo 6. Grafico De Pregunta 6 De Encuesta.	36
Anexo 7. Grafico De Pregunta 7 De Encuesta.	37
Anexo 8. Respuesta De Entrevista 1.....	37
Anexo 9. Respuesta De Entrevista 2.....	38
Anexo 10. Requerimientos Del Sistema.....	38
Anexo 11. Bitacora Del Proyecto.....	42

INTRODUCCIÓN

La tecnología actualmente está presente en el entorno de nuestras vidas, de forma tan desarrollada que es difícil de creer, pero no siempre fue así, es el caso del desarrollo de páginas web, en el cual en los inicios era más difícil de crearlas, pero esas dificultades crearon los cimientos para crear estándares, paradigmas y otras técnicas que ayudarán a mejorar lo que hoy se conoce en desarrollo.

El presente proyecto se refiere al desarrollo de una aplicación en el entorno de trabajo de desarrollo web Ruby on rails. Que se puede definir cómo uno de los entornos más productivos actualmente alcanzando a generar mayor utilidad a los desarrolladores por tiempo.

Características como su enfoque hacia la productividad y su naturaleza open source hacen que este framework con diversos recursos sea útil en distintos ámbitos del desarrollo web.

Para exponer distintos recursos y su desempeño se ha optado por implementar una cantidad de herramientas nativas y de terceros que una vez sean implementados y funcionales se prestarán para una mejor evaluación tanto a nivel práctica como técnico.

Utilizando la metodología de la investigación mixta, tanto cualitativa al agregar el análisis diagnóstico y las entrevistas para recolectar aspectos característicos del proyecto y cuantitativas para estimar las magnitudes de diversos criterios.

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

Desarrollo de un sistema web de registros en “ruby on rails” portando relaciones, validaciones, con capacidad para suplir necesidades básicas y prácticas de manera eficiente haciendo uso de herramientas en las cuales podamos realizar diversos tipos de evaluaciones en cada una de ellas.

1.1 Antecedentes

- El primer antecedente es sobre un trabajo elaborado en la Universidad técnica del norte (Ecuador) 2016, elaborado por Rubén Guerrero, quien expuso el tema; “Estudio comparativo de los frameworks “ruby on rails” y “django” para la implementación de un sistema informático de control y administración de network marketing”.

Se pretendía diagnosticar los recursos que se utilizaría en la implementación del sistema, tales como, diseño navegacional, interfaz intuitiva y uso del diseño hipermedio orientado a objeto, a lo que se concluyó mediante el análisis diagnóstico del rendimiento de estas herramientas que permiten generar mayor exactitud en cuanto a aspectos numéricos y una mayor eficiencia en la comunicación interna al igual que un ahorro de recursos como papel, tiempo, dinero y trabajo.

- El segundo antecedente es sobre el trabajo final de una maestría en ingeniería informática en la Universidad de las palmas de gran canaria (España) 2017, elaborado por la ingeniera Carolina Santana Martel, quien presento el tema; “Despliegue de una aplicación “ruby on rails” utilizando las tecnologías de virtualización Docker y CoreOS en la nube publica de Amazon Web Services”

Se plantea exponer una réplica practica de las tecnologías frecuentemente utilizadas por compañía en las cuales implementan el framework,

tecnologías como docker, coreOS y su desempeño orientado a implementación de contenedores simultáneos en test de pruebas entre otras funciones, lo cual se concluye con un alto y satisfactorio rendimiento, evaluado mediante tiempos de respuesta de ejecución, capacidad de ejecución simultánea y análisis técnico de su escalabilidad en la nube pública como lo es Amazon Web Services.

- El último antecedente es sobre el trabajo de monográfica en la Universidad Nacional Autónoma de Nicaragua - León (Nicaragua) 2018 presentado por los bachilleres; Lenin García y Jefferson Rojas, con el tema: “Elaboración de guías prácticas de laboratorio para el componente curricular Software como un Servicio de la carrera Ingeniería en Telemática del Departamento de Computación de la Universidad Nacional Autónoma de Nicaragua, León.”

Se pretende crear un formato de enseñanza con guías prácticas de desarrollo de software utilizando frameworks tales como “ruby on rails” como herramienta de implementación creando conocimiento sólido, practico y sintáctico sobre el lenguaje y el framework en cuestión, al finalizar el trabajo se crearon 9 guías y 2 proyectos finales los cuales mediante el análisis comparativo de las competencias que un estudiante debe alcanzar y la cualidades que aportan estas guías se termina por considerar que se alcanzaron los objetivos de forma satisfactoria.

1.2 Objetivos

Objetivo general

Proponer un proyecto de desarrollo web construido con “ruby on rails” implementando las herramientas de las que dispone el framework con una estructura “modelo, vista, controlador” en una aplicación web, y así exponer las facilidades que aporta este y su comunidad en el entorno técnico hacia el desarrollador.

Objetivos específicos

Determinar la utilidad de las herramientas nativas del framework, como también aquellos recursos desarrollados por terceros, mediante la adecuada implementación técnica de estas en los diferentes campos del proyecto, evidenciando su rendimiento.

Identificar de manera objetiva la eficiencia y eficacia de las herramientas proporcionadas por el entorno a través del análisis de datos teóricos y experiencias prácticas, mostrando así ventajas y desventajas propias del desarrollo con “ruby on rails”.

Desarrollar prototipos de informes técnicos, utilizando las herramientas y recursos del framework, para la demostración de las facilidades y bondades que presenta esta tecnología de creación de proyectos web.

1.3 Descripción del problema y Pregunta de Investigación

Se centraliza en un proyecto de aplicación web realizado con ruby on rails utilizando diversas herramientas y recursos con módulos básicos dedicado a registrar notas de cursos universitarios, haciendo uso de informes y reportes. A partir de esto se genera la siguiente pregunta de investigación:

¿Como debe ser un proyecto de desarrollo web construido con “ruby on rails” implementando las herramientas de las que dispone el framework con una estructura “modelo, vista, controlador” en una aplicación web, para así exponer las facilidades que aporta este y su comunidad en el entorno técnico hacia el desarrollador?

1.4 Justificación

Se considera que el conocimiento adquirido mediante la creación de un proyecto web desarrollado con “ruby on rails”, pueda brindar un entendimiento objetivo y sustancial para crear una posición de concordancia o confrontación respecto a las ventajas y desafíos que este framework pueda aportar en el entorno general del desarrollo web.

Se decide emplear un método de investigación mixto siendo un proyecto de implementación, la cual exige metodología cualitativa como información oficial de los autores del framework, criterio de especialistas, así como el propio análisis práctico y metodología cuantitativa como las valoraciones generales de la comunidad que utiliza estas aplicaciones.

Más allá de generar un conocimiento del entorno del framework, se pretende mostrar sus distinciones técnicas en el desarrollo web de forma comprensible como también exponer una alternativa con respecto a los framework más convencionales.

1.5 Limitaciones

Debido principalmente por el tiempo el mayor inconveniente se presenta en la creación de diversos módulos, que con correcta dedicación se estima poder implementar entre 5 o 6 módulos, lo cual satisface para la finalidad primordial del proyecto, pero deja por desear un mayor enriquecimiento de complejidad.

Falta de registros masivos, al tratarse de la creación un proyecto único y sin registros, aun si se procura crear la mayor cantidad de registros posibles, se volverá imposible contar con una base de datos voluminosa, lo cual vuelve los tiempos de carga de solicitud de datos imperceptible en la experiencia del usuario.

1.6. Variables

Variable Independiente

Proyecto de desarrollo web

Variables Independientes

Utilidad de las herramientas nativas del framework,

Eficiencia y eficacia de las herramientas

Prototipos de informes técnicos,

1.7.- Categorías, temas y patrones emergentes de la investigación

La necesidad del cambio ha generado nuevas tecnologías para desarrollar proyectos web cada vez más intuitivos. No hubo patrones emergentes en la investigación, además, los temas que genera esta indagación son de gran aporte para el sector tecnológico debido al análisis y resultado de los datos adquiridos.

1.8 Supuestos básicos

Se cree factible realizar un proyecto con suficientes implementaciones sobre las herramientas únicas del framework, como también de recursos de terceros implementados por un desarrollador junior en un tiempo estimado de 30 horas de aplicación.

Se considera alcanzable evidenciar la importancia de los beneficios que generan estos recursos de manera individual, al igual que el rol que desempeñan en conjunto para el completo entendimiento del funcionamiento sincrónico.

Se valora que mediante el correcto análisis técnico del conocimiento práctico adquirido y teórico podemos evaluar correctamente las deficiencias y fortalezas que aporta este entorno.

1.9. Contexto de la investigación

La temática sustancial que envuelve el desarrollo lógico del proyecto es la de crear una aplicación web capaz de ser ejecutarse en un servidor para distintos clientes con la finalidad de poder registrar un pequeño ecosistema de notas el cual incluya profesores, estudiantes, materias, cursos y notas de estos, a su vez que sea capaz de generar informes acerca de las notas y pueda generar reportes pdf de estas últimas.

En la presente aplicación se utilizará distintas librerías entre las cuales veremos: la librería de ransack para realizar las consultas de filtrado del informe de notas, se utilizará también las librerías “wkhtmltopdf” y “wicked_pdf” para poder generar el reporte pdf del informe, la librería “pg” para la interfaz de ruby hacia PostgreSQL y la librería de devise la cual se utilizará para la creación de usuarios.

Además de esto se utilizará herramientas originales del entorno, como lo son los generadores con los cuales se crean módulos básicos pero completos, formularios de preconfigurados y herramientas de validaciones.

Requerimientos específicos oficiales no existen el entorno de desarrollo se ejecuta de acuerdo a los recursos del servidor, sin embargo, como señala “Product Documentation” se recomienda como mínimo las siguientes especificaciones ya sea un servidor virtual o real.

- Por lo menos 2 GB de RAM
- Mínimo un procesador de 4 núcleos
- Un disco solido de 25 GB

Todo esto desarrollado en un servidor local Linux con el framework de rails con su

versión 6.1.0 y la versión 2.7.6 de ruby ejecutado en un equipo con sistema operativo Ubuntu versión 20.04, con procesador Intel(R) Core (TM) i5-4570 CPU @ 3.20GHz 3.20 GHz, con una RAM de 8 GB, y un disco duro de 250Gb.

El servidor remoto es el público de heroku, el cual se basa en limitaciones técnicas sobre su uso público, entre las más destacadas están; la solicitud HTTP tiene un tiempo de espera de 30 segundos para la respuesta inicial y un tiempo de espera de 55 segundos a partir de entonces. Máximo de 1 MB de búfer permitido para la respuesta, la clave de configuración y el par de valores están limitados a 32kb para una aplicación.

CAPÍTULO II: MARCO REFERENCIAL

2.1. Estado del arte

Es en el presente apartado en el cual se elabora el análisis y comprensión de información precedente, elaborado mediante proyectos similar al propio, que aportan material valioso. Debido que al tratarse de proyectos de aplicación web existen tantos como necesidades a cubrir, se ha optado por analizar un solo proyecto abordando sus aspectos de la manera más completa posible.

Proyecto objetivo

Se trata del trabajo de fin de máster en ingeniería informática el cual se titula “Despliegue de una aplicación Ruby on Rails utilizando las tecnologías de Virtualización Docker y CoreOS en la Nube pública de Amazon Web Services” de Carolina Santana Martel en la universidad de las palmas de gran canaria, 2017.

Este trabajo trata sobre el desarrollo y despliegue de una aplicación web construida con Ruby on rails, implementando herramientas prácticas propias del entorno del diseño a gran escala como el uso de contenedores en este caso docker y la utilización de una nube la cual para este trabajo se utiliza Amazon web service, entre otros recursos.

Objetivos

Implementar herramientas tecnológicas de virtualización tanto en el local como en la nube en Ruby on rails, mostrando así su flexibilidad, escalabilidad y robustez que éstas pueden ofrecer al desarrollo web.

Desarrollo

Se comenzó por la maquetación y realización del diseño de la base de datos, sus tablas y relaciones, luego se configuro el repositorio local para proceder a montar el diseño y crear los módulos y sus complementos, tales como diseños de vistas, validaciones, etc. Luego se incorporan test y la configuración necesaria para correr

Docker, lo cual incluye a CoreOS, para luego preparar el proyecto y exportarlo a la nube Amazon web service.

Una vez puesto en ejecución ambos entornos tanto local como remoto se desempeñaron de manera correcta con tiempos de respuestas esperados, al igual la forma de ejecutar los test en docker fueron totalmente éxitos.

En la etapa de actualización y mantenimiento también se realizaron sin inconvenientes, siendo un éxito su actualización y modificación del sistema.

Resultados y conclusiones

La aplicación ha sido configurada para implementar los servicios con posterioridad a la construcción de su imagen docker. Ésta vuelve a generarse con cada cambio o añadido en el Repositorio GitHub. Los objetivos específicos y generales definidos al inicio del proyecto se lograron. El estudio permitió conocer las tecnologías de virtualización, contenedores y orquestación implementadas, análisis de la aplicación en general en la cual se pudo conocer Las diferentes capas, características y componentes, así como obtener información práctica del uso de éstas y sobre las propuestas de su uso tanto local como en la nube.

2.2 Perspectiva teórica asumida

Aspecto epistemológico

Framework: Traducido literalmente del inglés como “marco de trabajo” se puede entender este concepto como un entorno de trabajo que proporciona herramientas previamente diseñadas para facilitar la resolución de una serie de problemas comúnmente presentes en la implementación de estos según sea el enfoque de estos entornos.

Ruby on rails: framework desarrollado en el lenguaje Ruby, diseñado especialmente para el rendimiento de la escritura de código y disminuyendo el tiempo dedicado a

las configuraciones que ya están establecidas mediante convenciones, basado en el paradigma MVC, Open Source y dirigido principalmente al desarrollo web.

Open source: Se hace referencia al software de código abierto, el cual significa que los derechos de autor están libres, es decir son gratuitos para el uso que se la comunidad les quiera dar, también se puede modificar y en ocasiones aportar recursos para el mantenimiento de este.

Aspecto filosófico

Dry: Del inglés “Dont repeted yourself” que traducido al español seria “No te repitas a ti mismo” es un paradigma bajo el cual se escribe código, el cual tiene como objetivo primordial no repetir aquel código que no sea estrictamente necesario, promoviendo así el compartir bloques de código que tengan una función específica y se pueda reutilizar en las distintas interfaces en las cuales se necesite emplear.

Convención: Este es otro paradigma bajo el cual se apega rails, el cual tiene como principio “convención sobre configuración” lo cual se basa en tener ciertas flexibilidades y contracciones al programar, que se dan por entendidas al relacionar interfaces, en post de agilizar la codificación.

Generadores: Son parte de las funciones principales ofrecidas por rails, su principal tarea es crear interfaces del sistema en menor o mayor medida, lo que genera código funcional y muchas veces relacionado entre diferentes clases, esto ayuda a agilizar la creación de bloques de código que finalmente terminaran de ser más complejos.

Gemas: Son librerías propias de terceros externas al framework “ruby on rails” inicial, pero que están pensadas para utilizarse en éste, estos bloques de código proporcionan un funcionamiento específico según sea la necesidad de su propósito e implementación. Dependiendo en cual clase se necesite implementar se manda a llamar para disponer de sus funciones.

Aspecto científico

MVC: Paradigma el cual contempla la creación de un sistema bajo la lógica de funcionamiento “modelo, vista, controlador” el cual indica que el modelo es el encargado de validar datos y almacenarlos, el controlador es el encargado de interpretarlos y comunicarlos a la vista que finalmente tiene el propósito de mostrarlos.

Librería o bibliotecas: Del inglés “library” se entiende librería o biblioteca a uno o más bloques de código que contiene funciones específicas para suplir una necesidad del funcionamiento de la aplicación.

Consultas SQL: Son aquellas que interactúan con la base de datos SQL, encargadas de ordenar el almacenamiento, la eliminación y las búsquedas entre otras, de los datos almacenados.

Validaciones: Es el proceso comúnmente encontrado en el modelo, que se encarga de verificar que la orden que se pretenda ejecutar no afecte la lógica con la cual se creó la aplicación y cumpla con todos los requisitos según sean los campos involucrados.

Servidor: Es el computador que se encarga de procesar todo lo relacionado con almacenar y proporcionar consultas e interfaces solicitadas por el usuario, al igual que es el encargado de correr validaciones y realizar cálculos.

Las aplicaciones web

Son programas que funcionan en internet. Es decir, que los datos o los archivos en los que trabajas son procesados y almacenados dentro de la web. Estas aplicaciones, por lo general, no necesitan ser instaladas en tu computador.

En cualquier momento, lugar y desde cualquier dispositivo puedes acceder a este servicio, solo necesitas una conexión a internet y tus datos de acceso, que, por lo general, son el nombre de usuario y contraseña.

¿Como se logra?

El concepto de aplicaciones web está relacionado con el almacenamiento en la nube. Toda la información se guarda de forma permanente en grandes servidores de internet y te envían mediante un sistema previamente diseñado y programado, a tu dispositivo o equipo, los datos que requieres en ese momento, dejando una copia temporal dentro de tu equipo.

Estos grandes servidores de internet que prestan el servicio de alojamiento están ubicados alrededor de todo el mundo, así hacen que el servicio prestado no sea tan costoso, o que sea gratuito, en la mayoría de los casos, y extremadamente seguro.

¿Que se logra?

Se posibilita realizar una mayor cantidad de trabajo con una mayor eficiencia y una reducción en el margen de error. Esto para ahorrar trabajo y maximizar los resultados de los usuarios finales.

Inicios de “ruby on rails”

Ruby on Rails es un entorno de trabajo con el fin de desarrollar aplicaciones web, escrito en el lenguaje de programación Ruby. Ruby es un lenguaje dinámico orientado a objetos. Ruby vio la luz en 1995 de la mano de Yukihiro “Matz” Matsumoto, pero no se popularizo hasta 2005 cuando David Heinemeier Hasson lanzo “ruby on rails”.

Ruby on rails

Rails es un marco de desarrollo de aplicaciones web escrito en el lenguaje de programación Ruby. Está diseñado para facilitar la programación de aplicaciones web haciendo suposiciones sobre lo que todo desarrollador necesita para empezar. Te permite escribir menos código mientras logra más que muchos otros lenguajes y entornos. Los desarrolladores experimentados de Rails también

informan que hace aplicaciones web desarrollo más divertido.

Es un entorno de trabajo completo, con presentación de plantillas HTML, actualización de bases de datos, envío y recepción de correos electrónicos, mantenimiento de páginas activas, almacenamiento de cargas en la nube y provisión de sólidas protecciones de seguridad para ataques comunes.

Velocidad de Ruby

En los inicios de rails cuando el framework y el lenguaje ruby se popularizo fue cuando se comenzó a percibir cierta lentitud en páginas creadas con ruby, estas dilaciones se le atribuyeron primordialmente al lenguaje, que si bien es correcto responsabilizarle cierta lentitud debido a que entre otras cosas aún se encontraba en versiones tales como la 1.8 y la 1.9 había factores también que agravaban este caso, factores que incluso hoy en día siguen afectando al lenguaje.

Exactamente en la actualidad la versión más reciente de ruby es la 3.1.2, y con ello trajo una gran mejora en el rendimiento, únicamente de la 2.0 a la 3.0 se triplico su eficiencia y factores como el rendimiento en los procesadores actuales han mejorado de forma exponencial.

Actualmente la mayor causa de la lentitud de estas páginas se debe a la flexibilidad que se le ofrece al programador, al ser un lenguaje con múltiples formas de lograr el mismo resultado muchas veces no se opta por el código más eficiente.

Recursos importantes de ruby on rails

Generadores

Son herramientas de ruby on rails que se utilizan para acelerar la creación de hasta un módulo, estos generadores pueden crear controladores, modelos o vistas, al igual les es posible crearlos todos relacionados entre sí e inclusive relacionarlos foráneamente con otro modulo, aun con el código muy completo que se genera es necesario siempre retocar lo generado, agregar diseño y estilo a las vistas y crear validaciones que no incluye el generador en el modelo entre otras.

Migraciones de base de datos

Las migraciones nos permiten hacer cambios sobre el esquema de la base de datos de forma iterativa y consistente. Una migración es un archivo que contiene instrucciones para modificar el esquema de la base de datos (crear tablas, agregar columnas, eliminar columnas, eliminar tablas, etc.).

Cuando creas un modelo desde la línea de comandos con el generador de rails, automáticamente se crea una migración con las instrucciones para crear la tabla. Sin embargo, también puedes crear migraciones para agregar, remover o cambiar columnas de una tabla, entre otros.

Convenciones

Parte de la misión de rails es terminar con pequeñas pero abundantes configuraciones y que esté en constante crecimiento de soluciones repetitivas que enfrentan los desarrolladores que crean sistemas de información para la web. Hay miles de decisiones de este tipo que solo deben tomarse una vez, y si alguien más puede hacerlo por el desarrollador, pues mucho mejor.

La transferencia de la configuración a la convención no solo nos libera de la discusión, también proporciona un campo exuberante para el crecimiento de abstracciones más profundas. Si tenemos la capacidad de usar la dependencia de

la clase Persona en la tabla de personas, entonces podemos usar la misma transformación para mostrar la asociación, declarada como has_many: personas, para buscar la clase Persona. La fortaleza de las buenas convenciones es que generan dividendos en una amplia gama de usos.

Pero además de las ganancias en la productividad para los expertos, las convenciones también reducen las barreras de entrada para los novatos. En rails, hay muchas convenciones que un novato ni siquiera necesita conocer, pero que simplemente puede beneficiarse de ellas solo por el hecho de ignorarlas. Es posible crear grandes aplicaciones sin saber por qué todo funciona de la manera en que funciona.

Eso no es posible si su entorno de trabajo es tan complejo como innumerables instrucciones de la configuración de dicho sistema y su nueva aplicación es una hoja de papel en blanco. Se necesita un esfuerzo inmenso para averiguar dónde y cómo empezar. La mitad de la batalla para ponerse en marcha es encontrar un hilo del cual tirar.

Helpers

Son métodos que pueden ser usados por las vistas, bloques de códigos reutilizables entre las vistas del proyecto. Cuando se tiene un código de vista que implementa cierta lógica y que se debe utilizar en varias vistas, es adecuado extraerlo a un helper. Así no se repite código en la vista y además se mantiene más limpio y legible.

Concerns

Son módulos que guardan código común reutilizable entre múltiples clases. Para que lleguen a ser concerns, se debe extender de ActiveSupport::Concern y la clase que quiera usar dicho concern debe hacer un Include NombreConcern.

CAPÍTULO III: DISEÑO METODOLÓGICO

3.1 tipo de investigación

3.1.1 Tipo de diseño de investigación

Se clasifica como una investigación experimental, debido a la naturaleza de la presente investigación la cual inicia a partir del desarrollo de un proyecto informático, manipulando, analizando y observando posibles respuestas del objeto de estudio práctico, con enfoque mixto al recoger información de ambos tipos.

3.1.2 Nivel de estudio

Se considera de estudio descriptivo, debido a que el nivel de profundidad en el estudio tiene una gran armonía con respecto al manifiesto de información, incluso cuando es un proyecto la acción principal del estudio es dar a conocer cualidades que no precisan de novedad.

3.1.3 Paradigma o enfoque de investigación

El enfoque de investigación es mixto, debido a los diversos puntos de abordaje desde el cual se estudiará el proyecto y su entorno, al igual que la diversidad de los instrumentos de recolección de datos.

3.2 Área de estudio

3.2.1 Ubicación, tamaño, institución

Se localiza la realización del proyecto en Managua, Nicaragua. Sin ser su implementación limitada por esta localización, sino más bien por las necesidades que este cubre y los conocimientos que a su vez son manifestados en el estudio, debido a que, al tratarse de un estudio tecnológico descentralizado de lo social, el entorno geográfico pasa a un segundo plano, siendo sus aspectos limitantes más técnicos, como puede ser el idioma, o los servidores.

La ubicación virtual del proyecto es un servidor público de heroku.com.

3.3 Unidades de análisis

Sabiéndose la población teórica como aquella población previamente estudiada, analizada y establecida como capacitada e ideal para resolver diferentes tipos de instrumentos de recolección de datos, se procede:

Entendiendo que el estudio de este proyecto requiere de conocimientos prácticos sobre el uso de este tipo de aplicación, y por otra parte también conocimientos técnicos para el desarrollo esta misma, las unidades de análisis se consideran necesario que sea de población teórica, para que tengan conocimiento técnico o practico, es decir sean idóneos para responder, según sea el tipo de instrumento de recolección de datos que se les implemente.

3.3.1 *Tamaño de población teórica*

El tamaño de la población teórica para el instrumento de recolección de datos encuesta será de 25 personas, mientras que para la entrevista únicamente se tomaran en cuenta a 2 personas.

3.4 Instrumentos de recolección de datos

3.4.1 Fuente de información primaria

Encuestas y entrevistas, instrumentos de recolección de datos en general.

Encuestas: Dedicadas primordialmente a la población teórica con conocimientos prácticos del uso de aplicaciones desarrolladas en rails, posteriormente experimentando las funcionalidades del proyecto desarrollado en cuestión, se espera respuestas sobre el uso práctico de la aplicación y su desempeño, enfocado en la calidad que recibe el usuario final.

Entrevistas: Dedicadas exclusivamente a la población teórica con conocimientos técnicos, con experiencia en desarrollo web y siendo especialistas en ruby on rails, se espera respuestas de carácter técnico sobre recursos utilizados en el proyecto, aclarando conceptos, aportando recomendaciones y demás aspectos generales.

3.4.2 Fuentes de información secundaria (publicaciones, sitios web y bases de datos)

Fuentes de información teórica

Forma parte fundamental para el desarrollo del proyecto, gracias a este tipo de fuente sabemos tiempos teóricos de respuesta en el framework, al igual que valoraciones y cantidad de veces que han sido implantados herramientas de terceros, al igual que el funcionamiento exacto del entorno y sus herramientas.

3.5 Confiabilidad y validez de los instrumentos

Se realizó una prueba piloto sobre las encuestas y entrevistas que se consideraban cubren con las necesidades a responder de la investigación de proyecto, luego de realizarlas se consultaron posibles defectos según el entendimiento de quienes rindieron estas pruebas, detectando de esta manera potenciales mejoras en la sintaxis y planteamiento de estas, las cuales se corrigieron manteniendo y potencializando la mejor respuesta posible de forma más legible.

3.6 Procesamiento de datos y análisis de la información

Del instrumento encuesta de recolección de datos realizado a 25 personas sobre su conocimiento se les instruyo: Responder según su experiencia y percepción sobre el proyecto previamente evaluado y su nivel de satisfacción percibido de este.

Los datos que esto arrojo son los siguientes:

FIGURA 1. PREGUNTA 1 DE ENCUESTA.

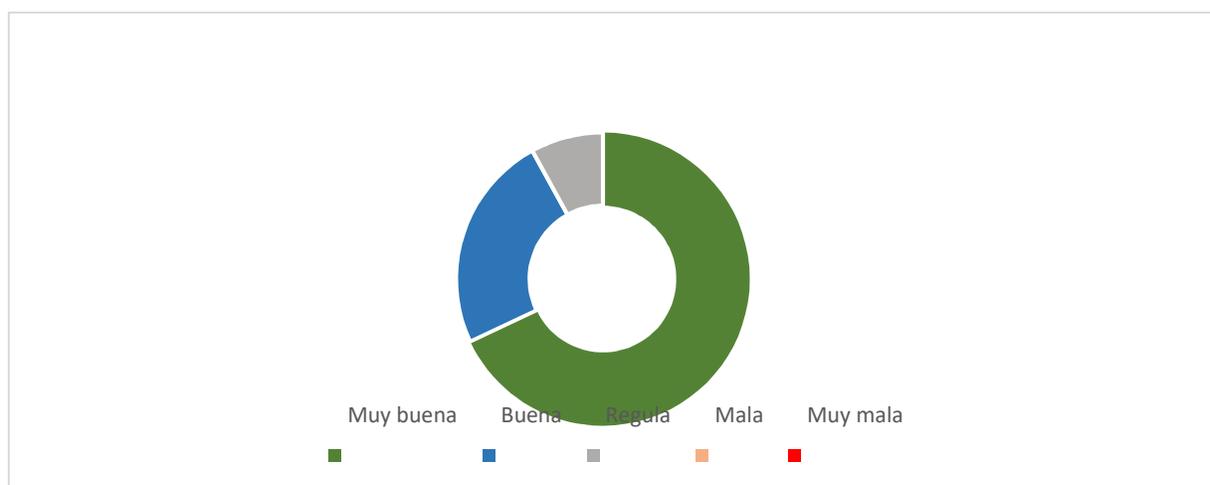
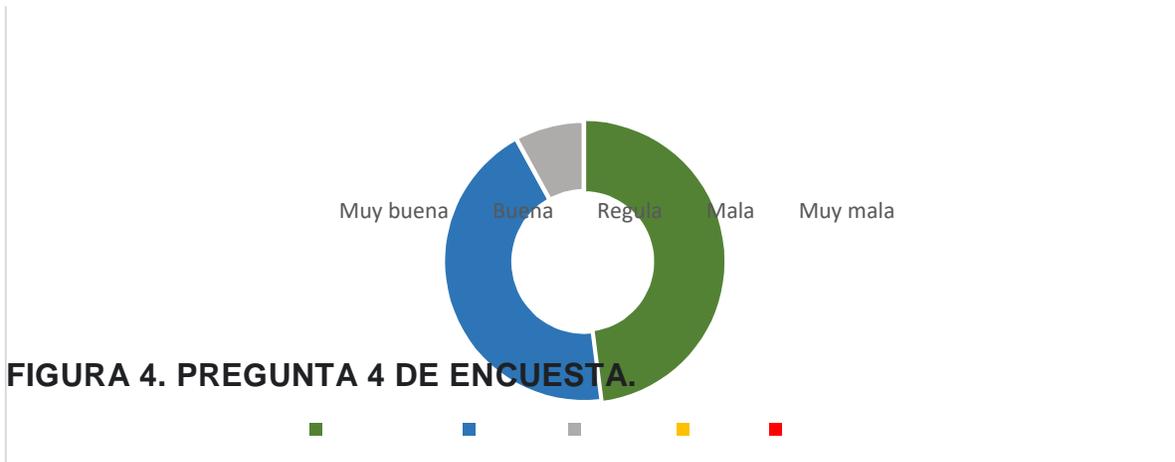
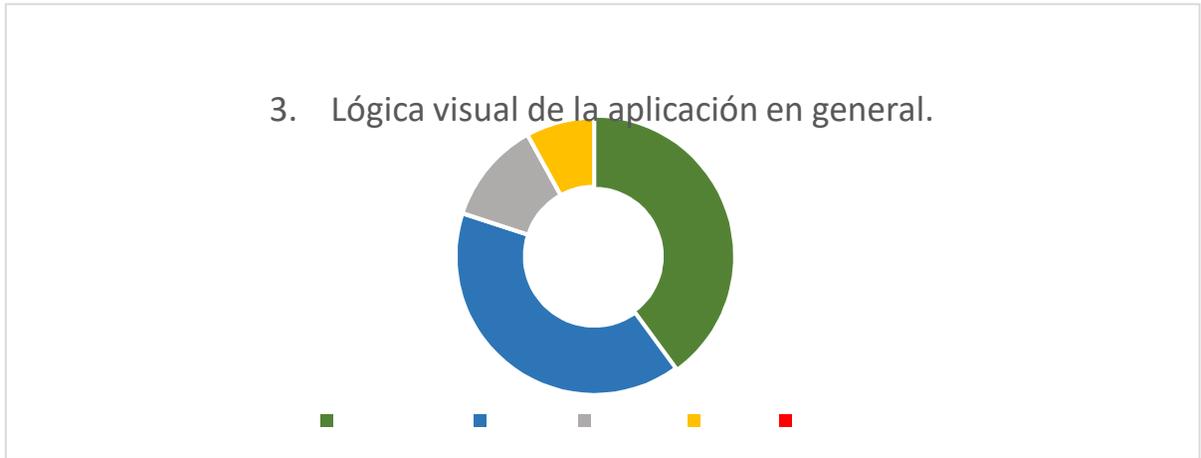


FIGURA 2. PREGUNTA 2 DE ENCUESTA.

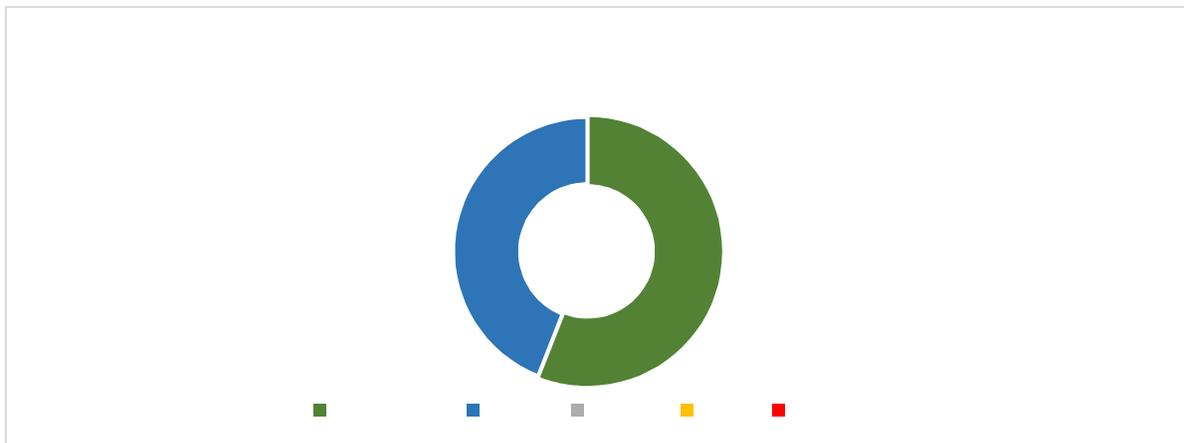
2. El correcto funcionamiento de la aplicación con respecto al informe y reportes pdf.

Muy buena Buena Regula Mala Muy mala

FIGURA 3. PREGUNTA 3 DE ENCUESTA.



4. Intuitividad en navegación de la aplicación en general.



Muy buena Buena Regula Mala Muy mala

FIGURA 5. PREGUNTA 5 DE ENCUESTA.

5. El correcto responder de la aplicación según la acción
previa realizada.

Muy buena Buena Regula Mala Muy mala

FIGURA 6. PREGUNTA 6 DE ENCUESTA.

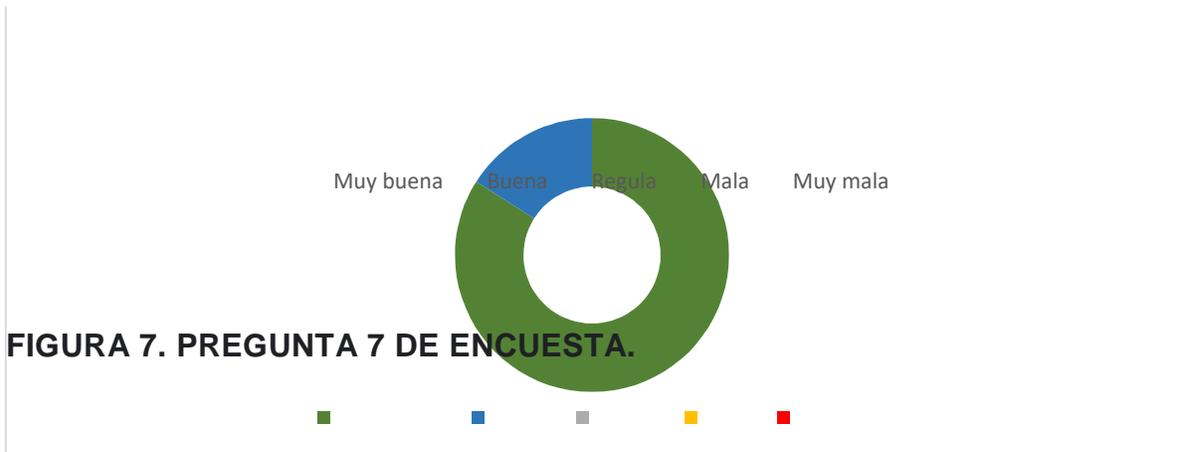
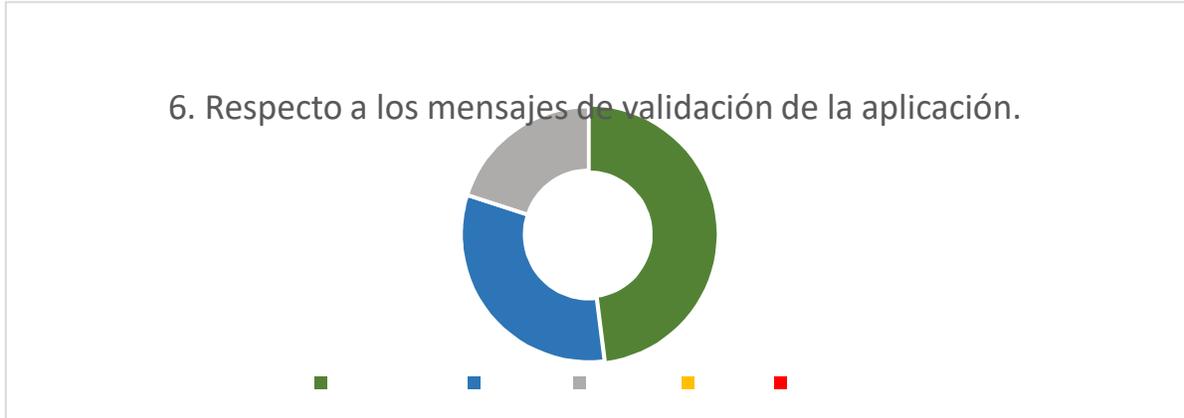


FIGURA 7. PREGUNTA 7 DE ENCUESTA.

7. El tiempo de respuesta de la aplicación al cargar, eliminar, guardar, mostrar o generar un registro.

Ver en anexos 1-7

Mientras que por parte de las entrevistas se obtuvieron respuestas muy concordes entre a los dos especialistas. En las cuales reflejaron y manifestaron lo siguiente:

La escalabilidad se vuelve realmente factible por parte del framework, sumándole el soporte que algunas librerías ofrecen para este cometido.

Librerías como ransack aportan una increíble adaptabilidad cuando se deben modificar tablas relacionadas de su uso y a su vez también cambiar sus parámetros, sin embargo, se debe tener en cuenta el correcto cambio en las relaciones de clases debido a que de estas relaciones se sirve ransack.

El entorno de rails es realmente flexible al momento de cambiar tablas con o sin relaciones, debido a sus migraciones este cometido puede no demorarse, pero se debe tener alto conocimiento sobre el diseño de estas para hacerlo correctamente.

Las librerías “wkhtmltopdf” y “wicked_pdf” ofrecen un amplio abanico de posibilidades desde renderizar a formato pdf informes y reportes de forma técnicamente presentables, hasta clonar la vista de html en un formato pdf, se recalca de igual manera la importancia de actualizar ambas librerías.

Los generadores nativos de rails que se encargan de crear gran parte de los módulos son de incalculable valor, debido primordialmente por rapidez, efectividad, seguridad y facilidad de uso.

La librería devise no solo es la mejor y más completa de las librerías referentes a registros de usuarios, sino también la más popular, ofrece una gran calidad en cuanto a seguridad y herramientas de soporte, también se caracteriza por tener un

alto mantenimiento.

Ver en anexos 8-9

3.7 Operacionalización de variables

Tabla 1. Operacionalización de Variables

Objetivo	Dimensión	Variables	Concepto de variables	Indicador	Instrumentos
Determinar la utilidad de las herramientas nativas del framework como también aquellos recursos desarrollados por terceros, mediante la adecuada implementación técnica de estas en los diferentes campos del proyecto, evidenciando su rendimiento.	Implementaciones nativas y de terceros	Flexibilidad	Habilidad de adaptación de las herramientas, según cambios dimensionales de la aplicación	Escalabilidad	Entrevista 1
		Utilidad	Efectividad de las herramientas según su respectivo uso	Modificable	Entrevista 3-4
		Requerimientos	Conocimientos y recursos tecnológicos necesarios para emplear estas herramientas.	Objetivo del recurso	Entrevista 5-6
Identificar de manera objetiva la eficiencia y eficacia de las herramientas proporcionadas por el entorno a través del	Rendimiento de recursos nativos y de terceros	Desempeño	Cumplimiento de la aplicación en sus distintos aspectos	Conocimientos	Análisis diagnóstico
		Calidad	Grado de comodidad según el usuario en	Correcta funcionalidad	Encuestas: 1-2
				Intuitividad	Encuestas: 3-4
				Interactividad	Encuestas: 5-6

análisis de datos teóricos y experiencias prácticas, mostrando así ventajas y desventajas propias del desarrollo con “ruby on rails”.			distintos campos de la aplicación.	Tiempo de respuesta	Encuestas: 7
			Requerimientos técnicos	Entorno ideal para ejecutar la aplicación, local y remotamente.	Servidores
Desarrollar prototipos de informes técnicos, utilizando las herramientas y recursos del framework, para la demostración de las facilidades y bondades que presenta esta tecnología de creación de proyectos web.	Correcta ejecución de informes.	Resultados prácticos	Efectividad para mostrar el informe según la petición solicitada.	Correcta funcionalidad	Encuesta: 2

Fuente: Elaboración propia

CAPÍTULO IV: ANÁLISIS DE RESULTADOS

En relación a la implementación de los recursos nativos y de terceros tal como se deja evidenciado en las entrevistas estos únicamente dejan reseñas positivas en cuanto a flexibilidad y utilidad, esto debido a su efectivo funcionamiento, afirmación que se pudo corroborar al implementarla de primera mano en nuestro proyecto, en el cual se implementaron sin mayores contratiempos. siendo destacable así también las observaciones que realizaron sobre el correcto uso de las buenas prácticas dado que existe cierta dependencia de estas para que todo funcione correctamente.

Referente al conocimiento que se implementó y se considera necesario para ejecutar el proyecto de forma correcta se pueden mencionar el manejo del lenguaje ruby, del entorno rails y de la configuración del servidor a un nivel de manejo de desarrollador jr.

Con respecto a la eficiencia y eficacia del rendimiento practico de las herramientas implementadas se pudo percibir mediante las encuestas realizadas una excelente satisfacción en relación a funciones básicas como guardar, mostrar y eliminar, como también al generar el informe y pdf del mismo, una satisfacción casi de igual magnitud se mantuvo en validaciones y respuestas del sistema y en correcto tiempo de respuesta que fue imperceptible, terminando por notar una pequeña disminución de aceptación en la lógica visual e intuitividad de la aplicación, aspecto más relacionado al diseño visual.

En referencia al entorno ideal de ejecución de la aplicación web tanto en el servidor local como el remoto siempre será lo mejor implementarlo en un servidor con las mayores prestaciones posible. Sin embargo, como se comentó en el marco contextual, se recomienda mantener un margen de prestaciones en estos servidores las cuales serían; Por lo menos 2 GB de RAM, mínimo un procesador de 4 núcleos y un disco solido de 25 GB.

CAPÍTULO V: CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN

A continuación, se presentan las conclusiones más importantes que se extraen del presente proyecto.

Los recursos nativos y los desarrollados por terceros son de gran utilidad para implementar funcionalidades que sean de gran rendimiento, de estos recursos existen tantos como aspectos de cualquier proyecto, estos muestran una gran calidad, flexibilidad y un alto desempeño generalizado.

Se pudo verificar la eficiencia y eficacia que genera los recursos de rails en la experiencia de utilizar una aplicación, tanto en campos básicos como en otros un poco más laboriosos, pero también se pudo conocer la estrecha relación que existe entre el éxito de estas herramientas y el correcto diseño en la lógica del desarrollo.

Se corrobora de forma específica la facilidad que ofrece rails para poder generar instrumentos como informes y reportes de forma efectiva.

Posibles líneas futuras del proyecto podrían ser, implementar roles y permisos a los usuarios, creando una seguridad completa y restrictiva, al igual que también podría ser colgar la aplicación en un servidor privado con dominio personalizado.

CAPÍTULO VI: RECOMENDACIONES

Para posibles proyectos de seguimiento o proyectos con similitud se realizan las siguientes recomendaciones.

Poseer asesoramiento de un desarrollador senior del entorno a implementar, muchas veces marca la diferencia entre progresar o seguir en la búsqueda de un camino.

Desarrollar un adecuado cronograma, teniendo en cuenta la efectividad propia con la cual se progresa.

Tener conocimientos previos sobre el proyecto en general, en todos los aspectos, aun si son muy básicos.

Mantener un alto grado de cuidado para que las Ubuntu es en las cuales se trabajan sean compatibles entre sí, esto ahorrara tiempo en actualizaciones cuando el proyecto ya este encaminado, evitando incompatibilidades y contratiempos.

REFERENCIAS BIBLIOGRÁFICAS

GCFGlobal (2022). *¿Qué son las aplicaciones web?*.

<https://edu.gcfglobal.org/es/informatica-basica/que-son-las-aplicaciones-web/1/>

Rails Guides. (s.f.). *Getting Started with Rails*. https://railsguides.es/getting_started.html

Ruby on Rails. (s.f.). *Compress the complexity of modern web apps*.

<https://rubyonrails.org/>

Make It Real Camp. (2022) Migraciones. Recuperado el 22 de julio de 2022, de

<https://makeitrealcamp.gitbook.io/ruby-on-rails-5/5-migraciones>

Ruby on Rails. (s.f.). *La doctrina Rails*. <https://rubyonrails.org/doctrine/es>

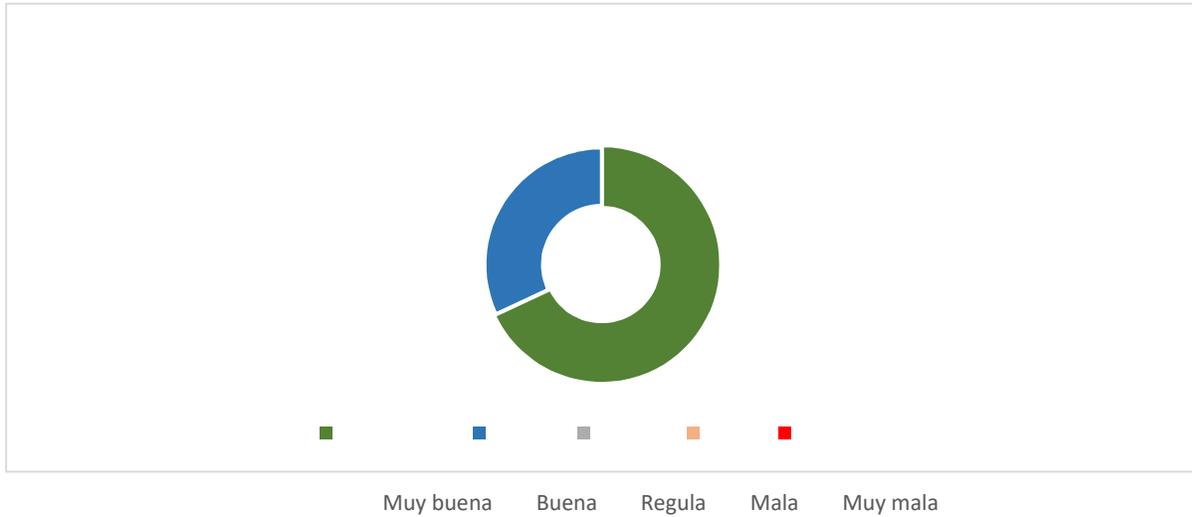
Mendivil, L. (2021). Concerns, Interactors y Helpers en Ruby on Rails [Entrada de blog]. Dudo, Investigo, Aplico y Comparto.

[http://lizzymendivil.blogspot.com/2021/04/concerns-interactors-y-helpers-en-ruby.html?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed:+DudoInvestigoAplicoYComparto+\(Dudo,+Investigo,+Aplico+y+Comparto.\)&m=1](http://lizzymendivil.blogspot.com/2021/04/concerns-interactors-y-helpers-en-ruby.html?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed:+DudoInvestigoAplicoYComparto+(Dudo,+Investigo,+Aplico+y+Comparto.)&m=1)

ANEXOS

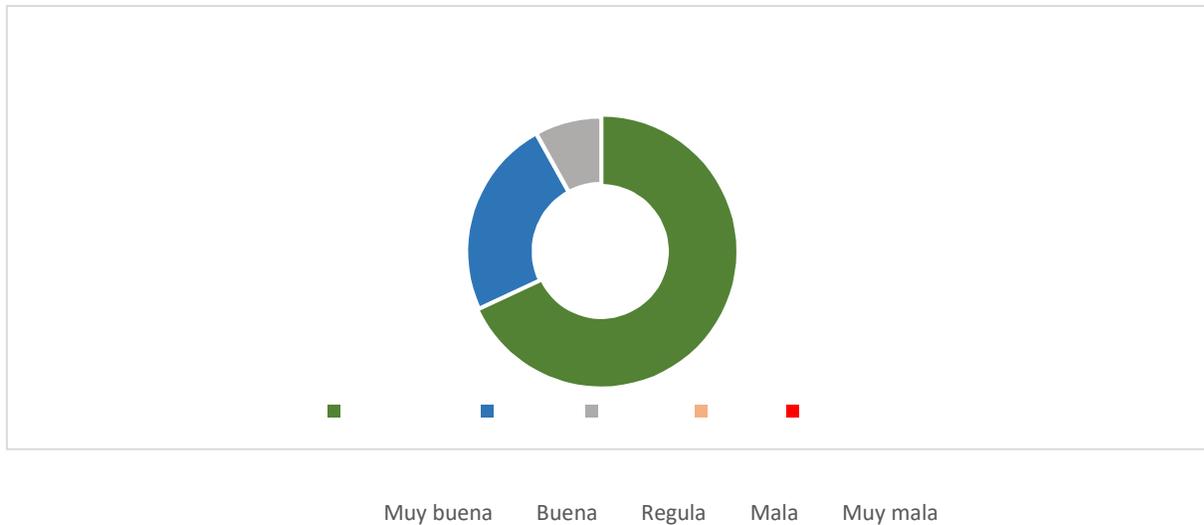
ANEXO 1. GRAFICO DE PREGUNTA 1 DE ENCUESTA.

1. El correcto funcionamiento de la aplicación con respecto a la vista, guardado y eliminado de los registros



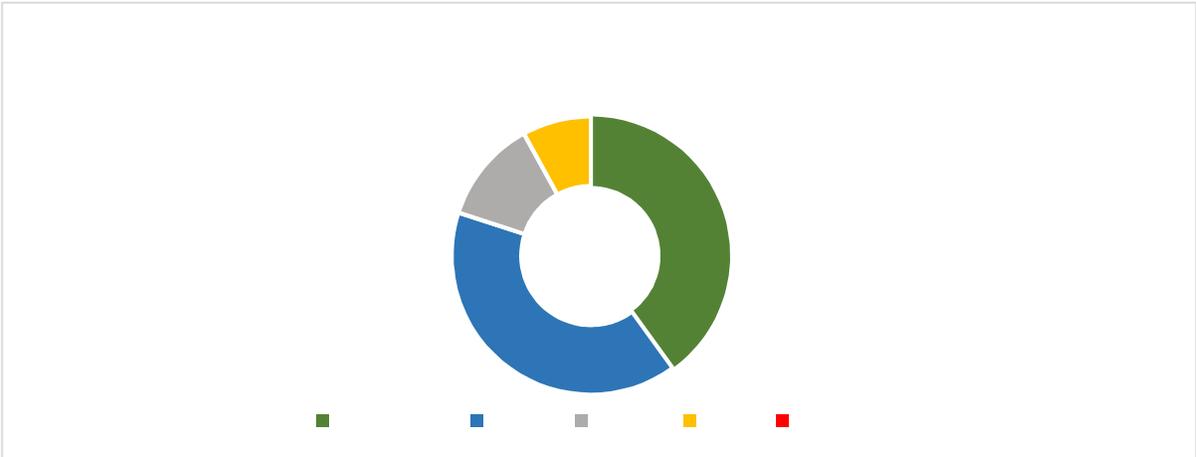
ANEXO 2. GRAFICO DE PREGUNTA 2 DE ENCUESTA.

2. El correcto funcionamiento de la aplicación con respecto al informe y reportes pdf.



ANEXO 3. GRAFICO DE PREGUNTA 3 DE ENCUESTA.

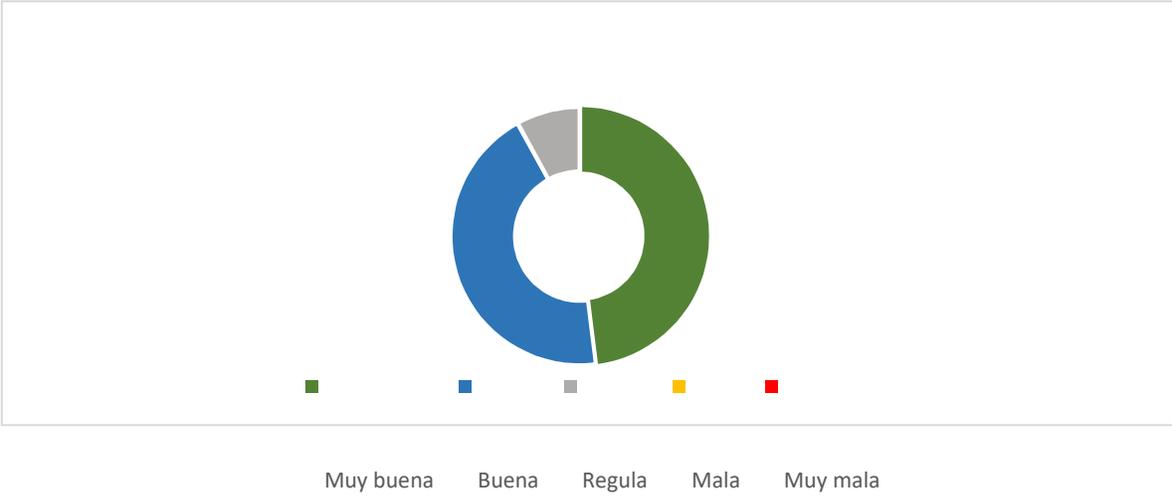
3. Lógica visual de la aplicación en general.



Muy buena Buena Regula Mala Muy mala

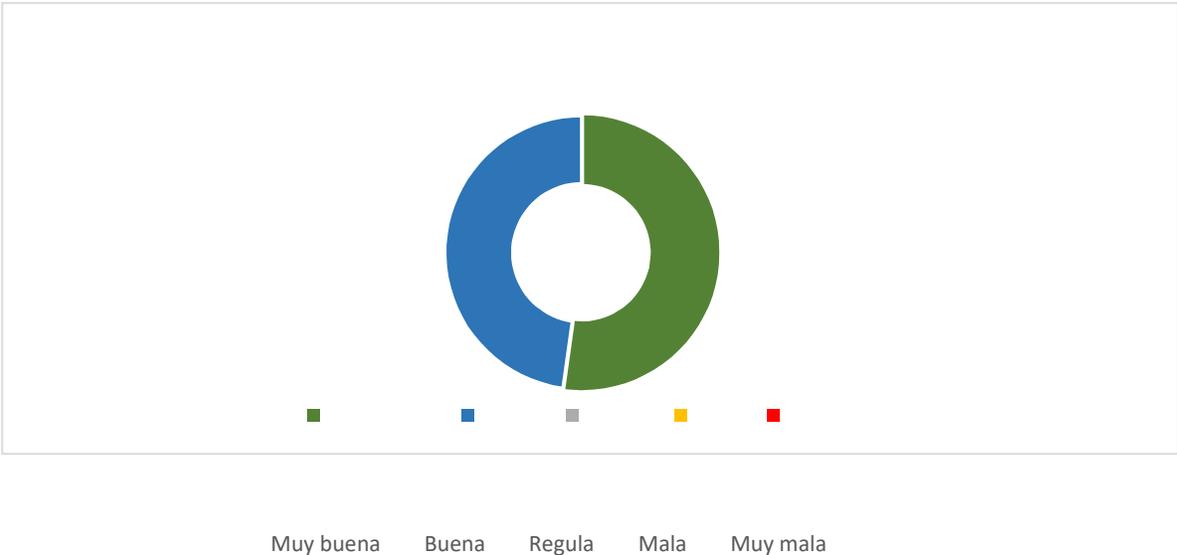
ANEXO 4. GRAFICO DE PREGUNTA 4 DE ENCUESTA.

4. Intuitividad en navegación de la aplicación en general.



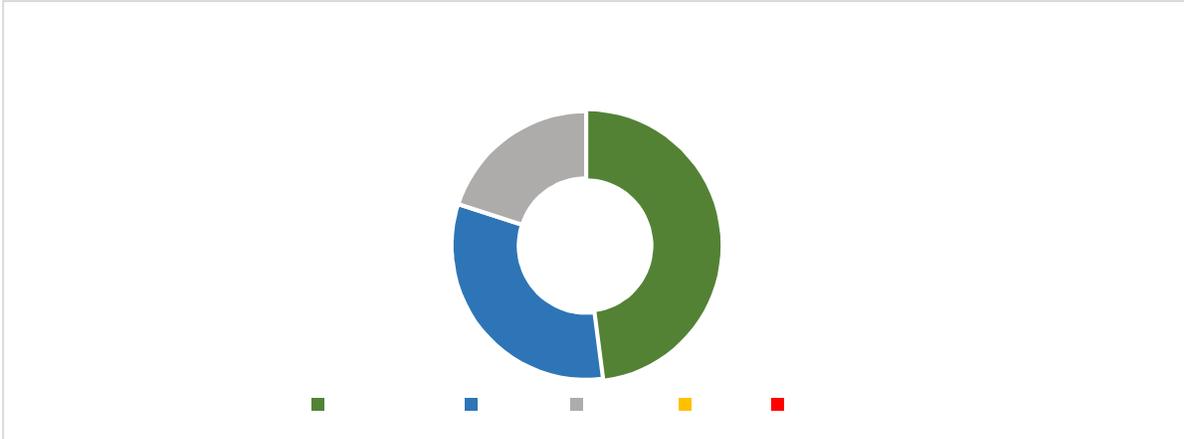
ANEXO 5. GRAFICO DE PREGUNTA 5 DE ENCUESTA.

5. El correcto responder de la aplicación según la acción previa realizada.



ANEXO 6. GRAFICO DE PREGUNTA 6 DE ENCUESTA.

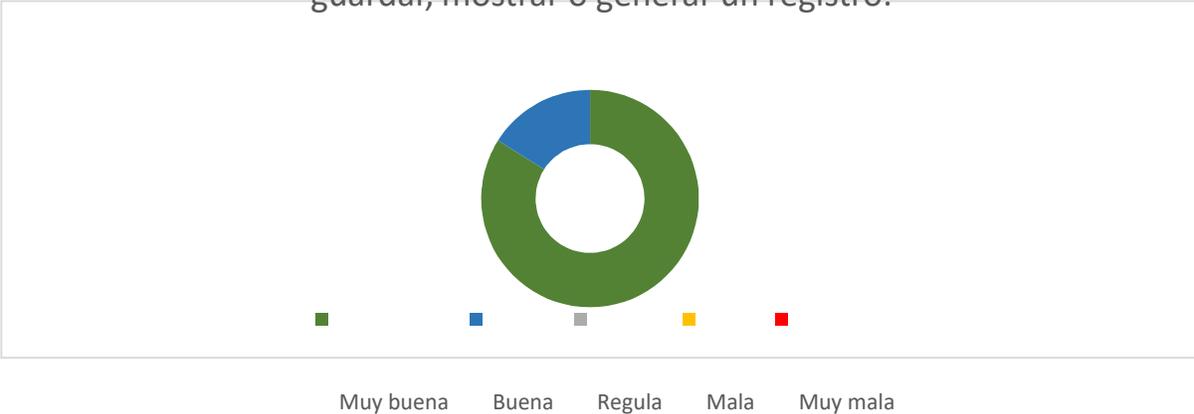
6. Respecto a los mensajes de validación de la aplicación.



Muy buena Buena Regula Mala Muy mala

ANEXO 7. GRAFICO DE PREGUNTA 7 DE ENCUESTA.

7. El tiempo de respuesta de la aplicación al cargar, eliminar, guardar, mostrar o generar un registro.



ANEXO 8. RESPUESTA DE ENTREVISTA 1.

Formulario de entrevista	
Responda de acuerdo a su criterio profesional	
*Obligatorio	
Nombre: *	Mauricio
1. ¿Qué tan escalable puede ser una aplicación desarrollada en ruby on rails? *	Muy escalable por todo lo que se puede hacer con las gemas.
2. ¿Qué tan bien responde a cambios tales como incrementos de parámetros la librería "ransack" de ruby on rails? *	responde bien, siempre y cuando estén muy bien realizadas las relaciones entre las clases
3. ¿Qué tan flexible considera una aplicación web de ruby on rails a cambios de tablas relacionadas? *	Considero que muy flexible
4. ¿Qué tan amplias son las posibilidades que ofrecen las librerías tales como "wkhtmltopdf" y "wicked_pdf" en ruby on rails? *	Muchas posibilidades, si se tiene muy actualizada la gema, ofrece varias opciones como diseño, gráficos, etc.
5. ¿Cuál es su opinión sobre la efectividad de los generadores de ruby on rails? *	Muy buena, rápidos, seguros y fáciles de usar.
6. ¿Cuál es su opinión sobre la efectividad de la librería "devise" de ruby on rails? *	Es la mejor para mi, es la mas utilizada por lo tanto es la mejor referente a seguridad mantenimiento etc.

ANEXO 9. RESPUESTA DE ENTREVISTA

2.

Formulario de entrevista	
Responda de acuerdo a su criterio profesional	
*Obligatorio	
Nombre: *	
Kender	
1. ¿Qué tan escalable puede ser una aplicación desarrollada en ruby on rails? *	
Bastante, todas las herramientas integradas son muy modificable si se mantiene una buena logica de desarrollo.	
2. ¿Qué tan bien responde a cambios tales como incrementos de parámetros la librería "ransack" de ruby on rails? *	
ransack tiene una gran habilidad de cambiar según los cambios que se realizen en los atributos.	
3. ¿Qué tan flexible considera una aplicación web de ruby on rails a cambios de tablas relacionadas? *	
Muy flexibles, Debido a las herramientas ofrecidas para este cometido.	
4. ¿Qué tan amplias son las posibilidades que ofrecen las librerías tales como "wkhtmltopdf" y "wicked_pdf" en ruby on rails? *	
Bastantes, gracias a estas se facilita la gestión de generar pdf y casi cualquier cualquier renderizado de html a este formato.	
5. ¿Cuál es su opinión sobre la efectividad de los generadores de ruby on rails? *	
En cualquier proyecto su implementación ahorra una gran cantidad de lógica y escritura de código, a demás su eficiencia es incuestionable.	
6. ¿Cuál es su opinión sobre la efectividad de la librería "devise" de ruby on rails? *	
Es muy buena frecuente se utiliza en proyectos de grandes magnitudes, además de su seguridad y recursos que ofrecen que es muy amplio.	

ANEXO 10. REQUERIMIENTOS DEL SISTEMA.

En el presente trabajo de desarrollo web, se realizará un sistema generalizado que cumpla con necesidades básicas de un entorno educativo para el registro y organización de datos universitarios enfocado hacia las notas de los estudiantes, con un ecosistema complementario.

Todo esto con la intención que pueda ser empleado de manera general en cualquier caso que cumpla con esta estructura, la cual se procura ser lo más estandarizada posible.

Objetivo principal

Dicho sistema deberá cumplir primordialmente con la funcionalidad de registrar a los estudiantes y sus notas, con datos relacionados tales como la oferta y el profesorado incluido en esta, agregando así información tal como direcciones y grados profesionales adecuadamente estructurada, esto con el fin de darle un mayor grado complementario.

Requerimientos

Funcionales:

- Que un limitado número de usuario puedan modificar todos los registros.
- Que los usuarios puedan modificar su usuario.
- Que exista un módulo de registro de profesorado que incluya datos tales como: nombre completo, cedula, profesión, correo, numero, dirección.
- Que exista un módulo de registro de estudiante que incluya datos tales

como: nombre completo, cedula, correo, numero, dirección.

- Que exista un módulo de registro para las asignaturas que incluya datos como nombre, créditos y descripción.
- Que exista un módulo de registro para los periodos que incluya datos como nombre, fecha de inicio, fecha final, año lectivo y descripción.
- Que exista un módulo de registro ofertas que contemple las asignaturas, periodos y profesores.
- Que exista un módulo de registro notas que contemple la oferta, el estudiante y su calificación.
- Que este el módulo de notas tenga un informe que pueda generar un reporte con un formato pdf.

No funcionales:

- Deberá ser maniobrable y reinventable en la medida de lo posible.
- Debe tener un alto grado de autonomía en relación con el mantenimiento técnico.

Requisitos

Requerimiento	Requisito	Resolución
Que un limitado número de usuario puedan modificar todos los registros.	Como estándar existirá 3 usuario modificables.	Se crearán 2 usuarios comunes y un usuario administrador. Pudiendo agregar mas

<p>Que los usuarios puedan modificar su usuario.</p>	<p>Los usuarios podrán modificar aspectos básicos del mismo siendo así el usuario administrador quien establecerá roles a todos.</p>	<p>Los usuarios modificarse a sí mismos en campos como la contraseña y el correo, y únicamente modificados en roles por el usuario administrador.</p> <p>Se creará un módulo roles el cual se le podrá asignar a más de un permiso.</p>
	<p>Se agregarán 3 módulos sobre la profesión los cuales serán: profesión, títulos y grado</p>	<p>Se crearán los módulos profesión, títulos y grado académico, todos tendrán su id, nombre y</p>
	<p>Se agregarán 4 módulos de dirección los cuales serán: país, departamento, municipio, ciudad.</p>	<p>Se crearán los módulos país, departamento, municipio, ciudad, todos tendrán su id, nombre y descripción y estarán relacionado de la siguiente manera:</p> <p>Registro departamento tendrá un país.</p> <p>Registro municipio tendrá</p>

<p>Que exista un módulo de registro de profesorado que incluya datos tales como: nombre completo, cedula, profesión, correo, numero, dirección.</p>	<p>Atributo nombre se dividirá en primer nombre, segundo nombre, primer apellido y segundo apellido.</p> <p>Profesor para especificar la dirección se relacionará con cada una de las tablas país, departamento, municipio, ciudad.</p>	<p>Se creará el módulo de profesor con los atributos del requerimiento de acuerdo a los requisitos.</p>
<p>Que exista un módulo de registro de estudiante que incluya datos tales como: nombre completo, cedula, correo, numero, dirección.</p>	<p>Atributo nombre se dividirá en primer nombre, segundo nombre, primer apellido y segundo apellido.</p> <p>Profesor para especificar la dirección se</p>	<p>Se creará el módulo de estudiante con los atributos del requerimiento de acuerdo a los requisitos.</p>
<p>Que exista un módulo de registro para las asignaturas que incluya datos como nombre,</p>		<p>Se creará el módulo de asignaturas con los atributos del requerimiento.</p>
<p>Que exista un módulo de registro para los periodos que incluya datos como nombre, fecha de inicio, fecha final, año lectivo y</p>		<p>Se creará el módulo de periodos con los atributos del requerimiento.</p>

Que exista un módulo de registro ofertas que contemple las asignaturas, periodos y	Se conformará por las relaciones de asignatura, periodo y profesores.	Se creará el módulo de ofertas con los atributos del requerimiento de acuerdo a los requisitos.
Que exista un módulo de registro notas que contemple la oferta, el estudiante y su	Se conformará por las relaciones de ofertas y estudiante, agregado a su atributo propio de	Se creará el módulo de notas con los atributos del requerimiento de acuerdo a los requisitos.
Que este el módulo de notas tenga un informe que pueda generar un reporte con un formato	Dicho informe se podrá filtrar de acuerdo al estudiante y la oferta. El reporte visualizara lo	Se creará una vista para el filtro y una para el informe, en esta última se mostrará la opción de
Deberá ser maniobrable y reinventable en la medida de lo posible.	Todos los registros serán modificables.	La asignación de roles permitirá al usuario administrador tener autonomía en el flujo de
Debe tener un alto grado de autonomía en relación con el mantenimiento técnico		Ha excepción de las actualizaciones y recuperación de cuenta de usuario el
	Para mayor consistencia en el control de los datos no se permitirá borrar ningún registro	Se creará la validación específica de no borrar registros relacionados que utilicen en otro

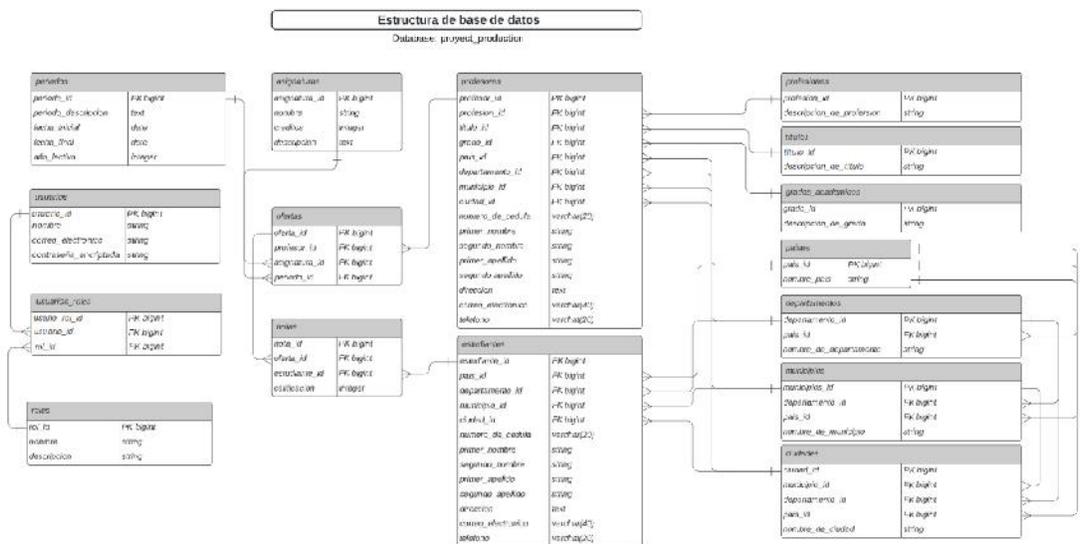
ANEXO 11. BITACORA DEL PROYECTO.

Antes de mencionar las descripciones técnicas del desarrollo que se realizó para llevar a cabo el proyecto, cabe especificar el entorno técnico, los cuales son; una computadora de 250 gb de disco duro, 8 gb de ram y un procesador i5 de cuarta

generación, con un sistema operativo Linux, específicamente Ubuntu versión 20.04, con la versión 2.7.6 de ruby y la versión 6.1.0 de rails.

- Se realiza el diseño estructural de la base de datos de la cual dependerá el resto del desarrollo. Tal como se observa en la figura.

Estructura de la base de datos.



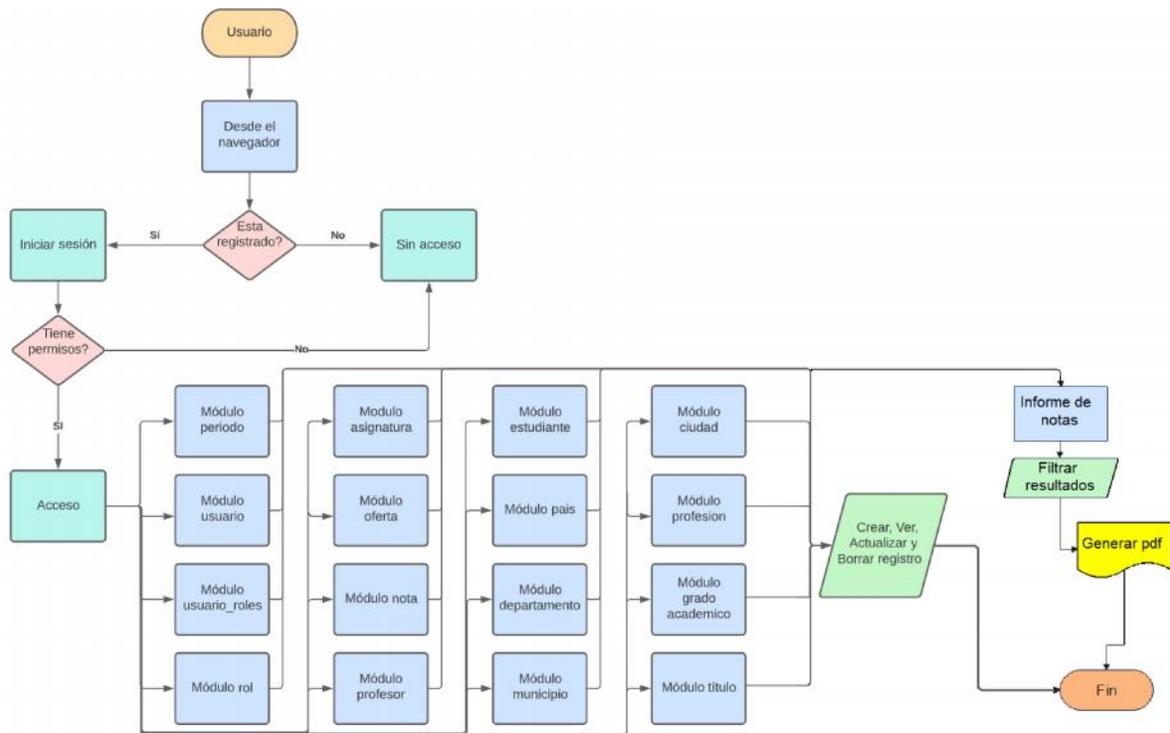
- Se crea el proyecto con el generador “rails new nombre_de_proyecto” el cual genera un entorno básico con la estructura modelo, vista, controlador. Carpetas dedicadas a estilo, gemas, concern, helpers entre otras.
- Se crean modelos, vistas y controladores básicos de las tablas: profesiones, títulos, grados_academicos, países, departamentos, municipios, ciudades, profesores, estudiantes, asignaturas, periodos,

ofertas y notas, todo esto con el generador scaffold para cada una de ellas, aplicando las relaciones para aquellas tablas que lo requieran, mencionando que estas relaciones quedaran en la base y no en el modelo, por lo que se tendrán que agregar a estos posteriormente.

- Se crea manualmente la vista de inicio de la aplicación y se configura como la vista predeterminada. En conjunto con todas las demás vistas se elige y aplica el diseño general de estas, haciéndolas a su vez dinámicas de acuerdo al tamaño de la pantalla, luego si es necesario se modifican los formularios estándares generados anteriormente por el scaffold.
- Se especifica en los controladores los parámetros que se utilizaran en la vista respectiva, esto para salvaguardar cualquier tipo de dato que pertenece al modelo, pero no debe visualizarse o modificarse desde este controlador, al igual que se especifica la variable que se mostrara en dependencia de la vista.
- Se realizan las vinculaciones en los modelos de acuerdo a lo establecido en la base de datos y se realizan validaciones requeridas.
- Se procede a crear el controlador y las vistas del informe de notas, realizando así el formulario con el cual se filtrará los registros que se requieran por medio de la librería “ransack”, la vista de registros, y la vista de estos mismos que se renderizará gracias a la librería “wicked_pdf” y “wkhtmltopdf” a pdf. Por parte del controlador se especifica el tipo de formato con el cual se responderá siendo así html o pdf según la petición solicitada.
- Se crea mediante el generador de la librería “devise” la tabla del usuario, el modelo, la vista y el controlador, todos estos elementos pudiendo ser modificados según la conveniencia, se modifica la vista al estilo general de la aplicación y se cambian los mensajes de validaciones a español.
- Se crean los modelos, vistas y controladores de roles y usuarios_roles mediante el generador scaffold, se establecen 5 permisos estandarizados con los cuales se podrán formar los roles, y de esa forma asignárselos a los usuarios en el módulo usuarios_roles.
- Dando por terminada la aplicación después de testear manualmente el

funcionamiento, se sube a heroku la aplicación y a github el código fuente.

Flujograma de la aplicación



ANEXO 12. Relación de autores

UNIVERSIDAD DE CIENCIAS COMERCIALES



UCC- SEDE MANAGUA

DIRECCIÓN DE INVESTIGACIÓN

Relación de Autores

Revisado por:

Lic. Fernando Monge- Dirección de
Investigación



Autorizado por:

Dra. Fabiola Somarriba – Vice Rectoría
Académica



*Por nuestro Prestigio, Trayectoria y Calidad
¡Somos la Universidad de la Gente que Triunfa!*