

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN



**Pontificia Universidad
Católica del Ecuador**

**“DISEÑO E IMPLEMENTACIÓN DEL PROCESO PARA LA GESTIÓN DE BOLSA DE EMPLEOS Y
PASANTÍAS BASADO EN EL CONSEJO DE EVALUACIÓN, ACREDITACIÓN Y ASEGURAMIENTO
DE LA CALIDAD DE LA EDUCACIÓN SUPERIOR PARA LA PUCE MATRIZ Y SUS SEDES, USANDO
LA METODOLOGÍA EXTREME PROGRAMMING”**

AUTOR:

SEBASTIAN EDUARDO ROBALINO ALTAMIRANO

DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS Y
COMPUTACIÓN

DIRECTOR:

MSC. FABIÁN DE LA CRUZ

Quito, 2019

Dedicatoria

El presente trabajo de titulación está dedicado a cada uno de los estudiantes que pertenecen a la Universidad Católica del Ecuador, que día a día se esfuerzan para cumplir sus metas y luchan con esmero anhelando convertirse en profesionales capaces de iniciarse en la vida laboral de nuestro país, cuyo sueño espero sea facilitado con este trabajo.

Agradecimiento

Agradezco a todas las personas que son parte de mi vida, aquellas que constituyeron inspiración y dolor, que ayudaron a formar mi carácter, y me llevaron a ser lo que soy, permitiéndome culminar la carrera que con tanto anhelo representa este trabajo de titulación.

Resumen

El presente trabajo de titulación trata sobre el desarrollo de un sistema que facilite la búsqueda de empleo a todos aquellos que ingresen sus datos al sistema, y que además se encuentren cursando o hayan cursado carreras en la Pontificia Universidad Católica del Ecuador, ofreciendo distintas posibilidades en el sector laboral con dos modalidades que son las practicas preprofesionales y empleo fijo.

Actualmente existe un proceso que sirve como bolsa de empleo, pero no es muy eficaz, por tanto lo que busca el sistema a implementar es automatizar el proceso actual, de manera que los usuarios puedan aplicar a tantas ofertas laborales como quieran, ya que estas son previamente filtradas para que solo aparezcan aquellas que solicitan específicamente el perfil del usuario en cuestión, pero ésta no es la única ventaja, además este sistema es capaz de recolectar toda la información generada y a partir de esta, crear indicadores que ayudan al mejoramiento continuo de la base de datos de la Dirección de General de Estudiantes, que es un sistema que dota con reportes e información necesaria a cada coordinador de cada unidad académica de la universidad.

Summary

This degree work is about the development of a system that facilitates the search for employment to all those who enter their data into the system, and who are also pursuing or have completed careers at the Pontifical Catholic University of Ecuador, offering different possibilities in the labor sector with two modalities that are pre-professional practices and permanent employment.

Currently there is a process that serves as an employment exchange, but it is not very effective, so what the system to implement is to automate the current process, so that users can apply to as many job offers as they want, since these are previously filtered so that only those that specifically request the profile of the user in question appear, but this is not the only advantage, in addition this system is able to collect all the information generated and from this, create indicators that help the continuous improvement of the database of the Directorate General of Students, which is a system that provides reports and necessary information to each coordinator of each academic unit of the university.

ÍNDICE

Capítulo 1 Análisis de los procesos Actuales	1
1.1. Situación actual de la Dirección General de Estudiantes	1
1.1.1. Misión.....	1
1.1.2. Visión.....	2
1.1.3. Objetivos	3
1.2. Análisis de los Procesos Actuales	3
1.2.1. Análisis de registro de Empresas	3
1.2.2. Análisis de aplicación a publicaciones	4
1.2.3. Difusión en redes Sociales	4
Capítulo 2 Marco teórico.....	5
2.1. Metodología de programación ágil Extreme Programming (XP)	5
2.1.1. Introducción	5
2.1.2. Características Fundamentales.....	6
2.1.3. Ventajas y desventajas	7
2.1.4. Adaptación y Aplicación de Extreme Programming al desarrollo Web.....	8
2.2. Lenguaje de programación del lado del servidor.....	8
2.2.1. Introducción	8
2.2.2. Características	9
2.2.3. Ventajas y desventajas	9
2.3. Lenguaje de programación del lado del cliente.....	9
2.3.1. Introducción	9
2.3.2. Características	10
2.3.3. Ventajas y desventajas	10
2.4. Servidor de plataforma NGINX	11
2.4.1. Introducción	11
2.4.2. Características	11
2.4.3. Ventajas y desventajas	11
2.4.4. Plataforma de gestión de bases de datos MySQL.....	12
2.4.4.1. Introducción	12
2.4.4.2. Características	12
2.5. Servicios REST.....	13
2.5.1. Introducción	13
2.5.2. Ventajas y desventajas	13
Capítulo 3 Selección de Herramientas.....	16
3.1. ¿Porque PHP?.....	16

3.2. ¿Porque servicios en REST?	16
3.3. ¿Porque base de datos relacional MySQL?.....	17
3.4. Arquitectura	18
3.5. Ciclo de vida	19
3.5.1. Fase de Exploración	19
3.5.2. Fase de Planificación en la entrega.....	19
3.5.3. Fase de desarrollo de Iteraciones	19
3.5.4. Fase de producción.....	20
3.5.5. Fase de mantenimiento.....	20
3.5.6. Fase de muerte del Proyecto.....	20
3.6. Costos.....	20
Capítulo 4 Sistema de Bolsa de Empleo y Pasantías PUCE	22
4.1. Administración de Roles	22
4.1.1. Estudiante	22
4.1.2. Coordinador	22
4.1.3. Reclutador de la Empresa	23
4.1.4. Administrador del Sistema	23
4.2. Ciclos de vida.....	23
4.2.1. Fase de Exploración	23
4.2.2. Fase de Planificación de la Entrega.....	24
4.2.3. Iteraciones.....	24
4.2.3.1. Hoja de Vida de los Estudiantes.....	24
4.2.3.2. Publicaciones de Empresas.....	25
4.2.3.3. Aplicación a Publicaciones	25
4.2.3.4. Reporte.....	25
4.2.4. Producción.....	25
4.2.4.1. Back-End	26
4.2.4.2. Front-End.....	28
4.2.5. Mantenimiento	28
4.2.6. Muerte del Proyecto.....	28
Capítulo 5 Conclusiones y Recomendaciones.....	30
5.1. Conclusiones	30
5.2. Recomendaciones	31
Bibliografía	42

ÍNDICE DE ILUSTRACIONES

Ilustración 1:Diagrama de arquitectura del sistema.....	18
Ilustración 2 Modelo entidad relaciones	27

ÍNDICE DE TABLAS

Tabla 1: Costos de servidores	21
Tabla 2. Herramientas	21

Capítulo 1 Análisis de los procesos Actuales

En este capítulo se describe los problemas y como utiliza el proyecto de Bolsa de Empleo en la PUCE, las deficiencias que tiene el proceso y como se podría mejorar con mejores prácticas y automatizando gran parte del proceso actual en un sistema online, alineado a la misión y visión de la PUCE para mantener el mejoramiento continuo, y renovando los roles actuales para administración del sistema y de los procesos.

1.1.Situación actual de la Dirección General de Estudiantes.

1.1.1.Misión.

a) "Es la única universidad católica que, de modo riguroso y crítico, contribuye a la tutela y desarrollo de la dignidad humana y de la herencia cultural mediante la investigación, la docencia y los diversos servicios ofrecidos a las comunidades locales, nacionales e internacionales.

b) Presta particular atención a las dimensiones éticas de todos los campos del saber y del actuar humano, tanto a nivel individual como social. En este marco, propugna el respeto a la dignidad y derechos de la persona humana y sus valores trascendentes, apoya y promueve la implantación de la justicia en todos los órdenes de la existencia, promueve la preservación del medio ambiente y el respeto a la vida.

c) Goza de la autonomía inherente a su condición de universidad, que le es necesaria para cumplir sus funciones eficazmente. Ejerce dicha autonomía con responsabilidad, y consiguientemente cumple con la rendición social de cuentas, tal y como lo determina la Ley.

d) Garantiza a sus miembros la libertad académica, salvaguardando los derechos de la persona y de la comunidad dentro de las exigencias de la verdad y del bien común.

e) Dirige su actividad hacia la formación integral del ser humano. Por ello trata de formar a sus miembros intelectual y éticamente para el servicio a la sociedad en el ejercicio profesional y en el compromiso con el desarrollo sustentable del país.

f) Pretende la integración del saber mediante el examen de la realidad con los métodos propios de cada disciplina académica y propiciando, al mismo tiempo, el diálogo entre estas para que se enriquezcan mutuamente.

g) Promueve el compromiso de todos los miembros de la comunidad universitaria para la consecución de los fines institucionales a través del diálogo y la participación, de conformidad con el presente Estatuto.

h) Como universidad particular ofrece una alternativa específica en el ámbito académico conforme a su propio Estatuto y reglamentos.

i) Como Universidad católica, se inspira en los principios cristianos; propugna la responsabilidad del ser humano ante Dios, el respeto a la dignidad y derechos de la persona humana y a sus valores trascendentales; apoya y promueve la implantación de la justicia en todos los órdenes de la existencia; propicia el diálogo de las diversas disciplinas con la fe, la reflexión sobre los grandes desafíos morales, religiosos y la praxis cristiana. Se han definido objetivos estratégicos para todos los ejes, como son: docencia, investigación, vinculación con la colectividad y administrativo, para cumplir con la misión descrita en el Estatuto institucional." (Planificación estratégica PUCE, 2016 - 2020).

1.1.2. Visión.

"La Pontificia Universidad Católica del Ecuador (PUCE), es el referente nacional en formación integral e inclusiva con impacto social. La innovación, agilidad y compromiso identifican su cultura organizacional. Es reconocida internacionalmente por su producción científica y la calidad de sus estudiantes y docentes." (Planificación estratégica PUCE, 2016 - 2020).

1.1.3.Objetivos.

- “Innovar e incrementar la oferta académica de grado y posgrado” (Planificación estratégica PUCE, 2016 - 2020)
- “Incrementar el número de graduados.”
- “Disponer de talento humano altamente calificado y cualificado”
- “Reposicionar el prestigio de la PUCE como una universidad de excelencia a nivel nacional e internacional”.
- “Aumentar el impacto de la PUCE en la mejora de la calidad de vida de las personas y comunidades”.
- ‘Promover la investigación y producción científica, con un enfoque innovador, para el desarrollo del conocimiento y la resolución de problemas globales y locales”.
- “Mejorar la gestión institucional asegurando la calidad y mejora continua”.
- “Administrar los recursos de manera sobria y eficiente”.
- “Buscar nuevas fuentes de financiamiento”.

(Planificación estratégica PUCE, 2016 - 2020)

1.2.Análisis de los Procesos Actuales.

1.2.1.Análisis de registro de Empresas en la dirección general de estudiantes.

Actualmente el procedimiento para realizar el registro de empresas en la Bolsa de Empleos PUCE es prácticamente nulo debido a que los requerimientos de cada empresa llegan primero a la unidad académica, por tanto, es el coordinador de cada una de estas unidades quien se encarga de procesar y validar si la oferta laboral cumple con todas las especificaciones para la aplicación de estudiantes. Posteriormente a la validación de dicha oferta, el coordinador realiza la impresión de la misma y la publica en la cartelera de la Facultad, o en su defecto accede a una base de datos en Excel personal que posee direcciones de correo electrónico de aquellos estudiantes que han tenido seguimiento de sus prácticas pre profesionales o que han acudido personalmente al coordinador, en el cual se basa para enviar e-mails de la oferta laboral en cuestión a aquellos estudiantes que el considere cumplen con las características requeridas por la empresa solicitante, ulteriormente a la difusión por medios públicos de la Universidad o el contacto vía email, el estudiante debe contactarse con la empresa demandante para concretar una entrevista. Este proceso no es del todo eficiente, ya que no arroja indicadores objetivos para realizar el reporte y seguimiento del mismo, además tampoco existe un método que nos permita asegurarnos llegar a todas las personas a quienes les pudiera interesar dicha oferta laboral

y muchas veces los posibles candidatos a una oferta laboral resultan perjudicados por el sistema de elección actual.

1.2.2. Análisis de aplicación a publicaciones.

En el presente es complicado para el coordinador de cada unidad académica reportar a la dirección de Bolsa de Empleos de la PUCE que estudiantes aplicaron a la oferta laboral publicada por él, debido a que para lograr acceder a esa información hoy en día el coordinador debe contactarse con las empresas que solicitaron el requerimiento y pedir toda la información pertinente. Esta labor es engorrosa y molesta tanto para el que solicita la información como para el que la da, inclusive esto podría ser un detonante para dejar de publicar sus ofertas laborales en la PUCE.

1.2.3. Difusión en redes Sociales.

El proceso de difusión actual es sesgado, esto se debe a que el mundo gira en torno a plataformas digitales, lo que produce que los medios físicos (como carteleras) ya no sean utilizados por los estudiantes, esta es la razón por la cual se ha tratado de usar redes sociales para la difusión de las ofertas laborales, pero es un poco complicado ya que este proceso se tiene que llevar a la par con el administrador de Facebook de la Bolsa de Empleos PUCE.

También existen otros medios de difusión como Alumni¹ cuyo proceso de publicación muchas veces no llega a la persona que administra esta plataforma y estadísticamente forma parte de las ofertas laborales que no llegaron a un estudiante que podía ser contratado o que cumplía perfectamente con el perfil de la misma.

¹ Alumni: En países anglófonos, el término alumni (plural de alumnus, el equivalente en latín para alumno) se refiere a los antiguos alumnos (exalumnos) o egresados de una institución académica

Capítulo 2 Marco teórico.

En este capítulo se detalla las características de la metodología elegida cuáles son sus ventajas las desventajas y sobre todo los beneficios que tiene frente a otras metodologías, también se describe los tipos de capas que se van a usar en el sistema y los tipos de ambientes que se necesita para poner en producción el sistema tomado de la mano con los servicios que se van a consumir y que se van a exponer para consumo en la web.

2.1. Metodología de programación ágil Extreme Programming (XP).

2.1.1. Introducción.

El desarrollo de software con el avance del tiempo ha evolucionado y con esto también la necesidad de documentar las reglas de negocio y lógica de funcionamiento de los sistemas, para tener una guía de mejoramiento continuo y pueda ser adaptable al pasar del tiempo y no se ate a ciertas tecnologías, sino que sea adaptable al tipo de ambiente en el que se ejecute el sistema.

El avance de los lenguajes de programación ha ayudado con la creación de las nuevas metodologías de programación y el progreso de las mismas, en la historia han existido varias propuestas de metodologías de desarrollo de software entre las principales tenemos:

- TSP: Team Software Process (Proceso de Equipo de Software).
- PSP: Personal Software Process (Proceso Personal de Software).

Los cuales aparecieron desde los 90's y han sido el medio para innovar las metodologías de Desarrollo de Software teniendo como eje principal la planificación detallada en lo más mínimo para que sean observables las funciones y los procesos del Software. Estas excelentes metodologías están sirviendo de inspiración para otras nuevas. Con el pasar del tiempo estas metodologías dejaron de ser usadas para proyectos emergentes y de corto tiempo porque se consideraba que estas metodologías demandan mucho tiempo en documentar el proceso más que en entenderlo, lo cual en cortos tiempos era contra productivo, a raíz de estos inconvenientes nacen las metodologías ágiles las que en la actualidad han tomado prioridad en los últimos años. Extreme Programming es una de estas metodologías que ha ganado peso con el tiempo, es considerada una de las metodologías más ligeras porque dan prioridad a las tareas que dan resultados directos y eficientes y reducen la burocracia en lo más mínimo pero registrando lo debidamente necesario dentro del ciclo de vida de la metodología.

2.1.2. Características Fundamentales.

Programación Extrema es una metodología ágil, que lo que busca es incrementar la interacción humana, es decir que para lograr un perfeccionamiento del desarrollo de software, necesita que exista trabajo en equipo, ya que XP se fundamenta en la realimentación continua entre usuarios y desarrolladores, por lo tanto la comunicación entre los participantes debe ser muy fluida, y las soluciones que se realicen deben ser simples, es por esta razón que esta metodología se utiliza esencialmente para proyectos que contengan exigencias indefinidas e inconstantes.

Dentro de las principales características de XP podemos observar que posee un sistema de desarrollo por cortos períodos de tiempo, es decir, se programa un lanzamiento del producto en 3 o 4 meses máximo, y dentro de este tiempo de lanzamiento se agendan varias iteraciones que durarán máximo de 2 a 3 días en las que se realizarán tareas pequeñas, y al final se implementarán en primer lugar las características que posean mayor prioridad, y aquellas que no fueron logradas en el plazo de tiempo requerido se les restará importancia.

La prioridad designada para cada característica se basará en 4 factores que son: costo, tiempo, calidad y alcance; y se determinará conjuntamente entre el equipo de desarrolladores y el cliente, es por esto que para que este sistema funcione adecuadamente el cliente debe estar involucrado completamente con el proyecto, además de disponer de una cantidad enorme de tiempo, puesto que va a ser él quien propondrá nuevas ideas en una primera instancia, y aprobará o sugerirá los cambios que considere necesarios en cada lanzamiento realizado. (Joscowikz, 2008) Estos cambios o ideas se generarán a partir de historias, éstas indican como debería funcionar el sistema, pero cada historia describe una sola cosa que el sistema tiene que hacer, y se debe poder probar, para que los desarrolladores puedan tener una idea correcta acerca del grado de complejidad y de cuánto tiempo demorará realizar dichas pruebas, por tanto el cliente tiene el poder de decidir el valor del negocio, que hacer primero, que posponer y que pruebas se mostrarán en el sistema en funcionamiento.

Otra de las características primordiales de este método es la simplicidad, es decir, a cada historia generada se debe asignar la solución que se pueda ejecutar más rápidamente durante cierta fase del proyecto, además estas soluciones deben: superar exitosamente cada una de las pruebas a las que se someta, no poseer lógica duplicada, implementar

exactamente lo que los programadores deseen y aplicar una mínima cantidad de métodos y clases.

Asimismo, como se había mencionado anteriormente XP promueve el trabajo en equipo para lo cual asegura que es mejor trabajar en pares de programadores, es decir dos desarrolladores en una misma computadora, a pesar de que se piense que esto duplicaría el costo de producción y ralentizaría el trabajo, en la realidad no es así, puesto que, al tener dos personas encargadas del mismo código, éste será revisado constantemente traduciéndose en un producto final mucho más refinado. Esta estrategia también es buena porque al final del proyecto hay más personas que conocen el código, siendo capaces de resolver problemas más fácilmente. Por último, este trabajo en conjunto promueve lazos sociales que permitirán a los desarrolladores trabajar más a gusto.

2.1.3. Ventajas y desventajas.

Como eje principal en una metodología de desarrollo de software es importante entender lo que el software debe hacer, cumpliendo las necesidades del cliente. A pesar de esto, lo que XP propone es un ciclo de vida dinámico en este caso es precisamente lo que se está buscando porque muchas de las veces el cliente no sabe lo que necesita y con esta metodología la incapacidad que tiene el cliente de especificar sus requerimientos al inicio del proyecto queda inhibida por sus ciclos de desarrollo cortos y dinámicos. Al acabar una iteración cada una tiene un entregable la cual hace palpable a cliente el avance del proyecto y que pueda proponer cambios o mejoras para el siguiente ciclo y así moldear al diseño ideal con una metodología que soporta cambios y no es engorrosa en la documentación, XP también tiene sus desventajas es super adaptable para proyectos cortos pero en proyectos largos no es recomendable usarlo porque no tienes revisiones de código o inspecciones también no soportar equipos distribuidos virtualmente.

Una de las mayores ventajas de este método es que al acabar una iteración cada una tiene una fase de entrega, que hace que los avances puedan ser palpables para el cliente y de esta manera pueda proponer cambios o mejoras para el siguiente ciclo y así ir moldeando el diseño ideal. Para que estas entregas se puedan realizar a tiempo se utiliza el código colectivo, es decir, todos son dueños del código, lo que permite que cualquier programador pueda arreglar partes del código que no se encuentren funcionando correctamente, evitándose así, los largos períodos de tiempo de espera hasta que el otro programador

reescriba el código, y corriendo el riesgo de que aun así no haya entendido lo que el primero quiso decir debiendo repetir el ciclo de espera en incontables ocasiones.

Este método también es concreto y existe una retroalimentación constante entre el cliente y los desarrolladores, por lo tanto, se planifica gradualmente, y este plan puede cambiar de acuerdo con las necesidades del negocio que vayan surgiendo. Además, como es sometido a pruebas constantemente, es más sencillo evolucionar y detectar errores en etapas tempranas, que posteriormente se solucionarán de la manera más sencilla posible, es decir para poder manejar este método no es necesario que los programadores posean habilidades extraordinarias.

En cuanto a las desventajas que presenta este método, podemos decir que principalmente está la comunicación constante necesaria entre el equipo de desarrolladores y el cliente, ya que éste no siempre va a tener la disponibilidad de tiempo que se requiere para la elaboración fluida del producto final. Además, XP está dirigido a grupos pequeños de no más de 20 participantes. Por otro lado, aún existen vacíos en el conocimiento sobre este método.

2.1.4. Adaptación y Aplicación de Extreme Programming al desarrollo Web.

La libertad, la conveniencia y la flexibilidad del acceso a plataformas Web han conducido a la proliferación de aplicaciones móviles, las prácticas de XP se configuran para mitigar los riesgos del proyecto (requisitos cambiantes dinámicamente, nuevo sistema debido a una línea de tiempo específica) y aumentar la probabilidad de éxito. El uso de XP ha sido sugerido para un desarrollo rápido de aplicaciones.

2.2. Lenguaje de programación del lado del servidor.

2.2.1. Introducción.

Es un lenguaje que está alojado en el servidor, donde se ejecuta los procesos pesados y la lógica de negocio del software, los archivos guardados y la data del software donde el cliente solo tiene acceso mediante restricciones y con limitaciones de la lógica del negocio a funciones del software mas no al código.

Hypertext Preprocessor (Preprocesador de Hipertexto) definición de PHP², es un lenguaje de programación de la década de los 90's fue uno de los primeros lenguajes de programación para generar HTML³ dinámico, también es conocido por ser el lenguaje más usado en la WWW⁴ y sigue estando en desarrollo por la comunidad de software libre de PHP.

2.2.2.Características.

La principal ventaja de PHP es que es un lenguaje muy usado y por esto hay documentación abundante de todas las versiones existentes y sobre todo no es cualquier tipo de lenguaje, sino que tiene su estructura, es orientado a objetos así que puede tener la estructura que el proyecto necesite para que adquiera la robustez que necesita.

2.2.3.Ventajas y desventajas.

Al ser el lenguaje más popular en la web podemos observar que cuenta con un sin número de ventajas, por ejemplo el costo; un servidor PHP básico cuesta entre 40 a 50 dólares anuales, el cual es un costo muy accesible en comparación con otros servidores, por lo tanto, es de suponer que la mayoría se incline por este lenguaje, ya que como sabemos, al momento de montar un servidor de producción hay que tener en cuenta el costo de mantenimiento y operabilidad de cada tipo de lenguaje. Otra ventaja fundamental es que al ser ilustre existen diferentes frameworks⁵ que fortalecen el lenguaje haciéndolo más fácil de conocer de implementar y de programar haciéndolo compatible con la mayoría de las bases de datos relacionales y no relacionales, por no decir con todas las bases de datos. Por un lado, una desventaja es que para correr código PHP se necesita un servidor para interpretar el código, al compilar se ejecuta el archivo, mas no genera un archivo compilado o dlls⁶ como en otros lenguajes.

2.3.Lenguaje de programación del lado del cliente.

2.3.1.Introducción.

El lenguaje de cliente en aplicaciones WEB es más conocido como front-end que prácticamente es CSS y HTML dinámico con Javascript, siempre está en constante cambio

² PHP: Acrónimo recursivo en inglés de Hyper text Preprocessor

³ HTML: Acrónimo recursivo en inglés de Hyper Text Markup Language

⁴ WWW: Acrónimo recursivo en inglés de World Wide Web

⁵ Frameworks: es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar

⁶ DLLS: Acrónimo recursivo de inglés de Dynamic-link library

con nuevas librerías y nuevos frameworks e incluso nuevos sub lenguajes. Hoy en día JS (JavaScript) ha tomado mucha fuerza siendo el lenguaje más popular para desarrollar apps móviles multiplataforma y más aún en aplicaciones web. Siendo el Sistema de Bolsa de Empleo un sistema multicapa está considerado que el sistema tenga funcionalidades asincrónicas donde se explotan las funcionalidades JS como las peticiones en tiempo real a la base de datos comúnmente llamadas peticiones AJAX.

El desafío de usar tecnologías multicapa es nivelar la carga de procesos y que no abarque funciones de alto rendimiento el front-end sino que sea una experiencia totalmente transparente para el usuario.

2.3.2.Características.

El código de lado de cliente puede estar compuesta por elementos de navegación imágenes menús, listas, etc. y un contenido principal gestionado por un componente. Los componentes son los niveles intermedios entre el back-end y el usuario, y son los que manipulan los contenidos para presentarlos de una u otra manera. Un componente puede ser algo así como una pequeña aplicación independiente que se integra en la aplicación para realizar una función, como listar artículos o una galería de imágenes. El componente de listas se encarga de realizar cosas con cada elemento de la lista, por ejemplo, puede mostrar una interacción con cada elemento de la lista, o un elemento concreto, o un formulario para que los usuarios envíen artículos. Estas distintas modalidades o expresiones del componente se denominan componentes hijo y la idea de esto es crear cuantos componentes sean necesarios para que funcione la aplicación. Mientras la aplicación se derive en más componentes será más robusta.

2.3.3.Ventajas y desventajas.

Si bien es cierto JS es el lenguaje de front-end más popular y usado para el desarrollo de aplicaciones WEB en este proyecto se usará JS con jQuery para cada componente AJAX del sistema. jQuery siendo una librería de compilación sobre el navegador la ventaja y desventaja más relevante es que al inspeccionar el código podemos ver la lógica de programación ahí, sin necesidad de artilugios o de programas avanzados solo con el intérprete del navegador, es bueno porque nos ayuda a entender la lógica de los demás programadores, pero también es malo para nosotros porque exponemos nuestra lógica a toda la WWW. Esta es la razón principal por la que se hacen las operaciones de base de datos

y algoritmos de la lógica de negocio en el back-end usado solo el front-end en lo más mínimo para hacer dinámico la interacción del sistema con el código HTML.

2.4. Servidor de plataforma NGINX.

2.4.1. Introducción.

NGINX es un servidor web que ofrece un excelente rendimiento, debido a su estructura asíncrona es decir que entrega la información que se solicita y basada en eventos es capaz de ofrecer contenido en formato MP4 y FLV. Además de ser muy fluido en la entrega de datos, soporta un gran número de conexiones simultáneas lo que lo hace ideal para permitir el uso al máximo de la estructura con la que esté interactuando.

Soporta muchos de los protocolos web "back-end" como WSGI, FCGI Y CGI, es por tanto una pieza estable del software que ha estado en funcionamiento por más de una década y ha construido un grupo bastante fuerte de desarrolladores a su alrededor ya que permite una variedad inmensa de opciones de configuración y es muy sencillo de usar.

2.4.2. Características.

En primer lugar, está la rapidez; es muchísimo más veloz que sus competidores debido a que tiene bóvedas asíncronas, es decir que no recarga cada vez que se le da una orden, y puede manejar miles de conexiones con un proceso por núcleo, permitiendo que tanto la carga del CPU como el consumo de memoria sea mucho más ligero. Además, es fácil de usar.

Con este servidor web, leer y retocar los archivos es mucho más sencillo que con otros servidores. Igualmente está la "modularidad", ya que viene con una gran cantidad de módulos distribuidos en el archivo original, y muchos otros se pueden descargar online.

2.4.3. Ventajas y desventajas.

NGINX y Apache son servidores web, realizando una comparación de sus últimas versiones, se observa una competencia muy cerrada en la mayoría de las áreas. Para el contenido estático, NGINX es el rey, pero en cuanto a contenido dinámico, la diferencia de rendimiento entre estos dos es bastante pequeña. Además, NGINX brilla con algunas de sus características más avanzadas, así como su soporte comercial y capacitación.

Por otro lado, los usuarios de alojamiento compartido pueden preferir la conveniencia del archivo .htaccess de Apache, ya que éste es mejor al momento de admitir la carga de varios módulos dinámicos, una característica que NGINX recientemente agregó. NGINX se utiliza principalmente para alojamiento de VPS, alojamiento dedicado o contenedores de clústeres.

Los propietarios de sitios web de alto tráfico que necesitan brindar una gran cantidad de contenido estático y / o transmisiones de medios probablemente preferirán NGINX. En la mayoría de los casos de uso de sitios web, cualquiera de los dos servidores web hará bien el trabajo. Para comenzar con cualquiera de los servidores web, se necesitará un proveedor de alojamiento de Linux sólido. Afortunadamente, en nuestro caso el área de informática de la PUCE solicita que el servidor sea Linux para alojar todos aquellos servicios que requiera el sistema.

2.4.4. Plataforma de gestión de bases de datos MySQL.

2.4.4.1. Introducción.

MySQL se puede usar para una variedad de fines, puede ayudar a equilibrar la carga del servidor en múltiples máquinas, facilitar las copias de seguridad, proporcionar una solución a la falta de capacidades de búsqueda de texto completo en InnoDB. Aunque MySQL tiene una reputación de robustez y pérdida de datos es un problema con muy pocas incidencias, es mejor estar preparado para cuando los datos se corrompan o se pierdan. La experiencia de usuarios demuestra que eventualmente sucederá y probablemente cuando menos lo esperamos. MySQL tiene licencia propietaria de Oracle pero para este sistema se utilizará el mismo lenguaje de MySQL pero en una base de datos de código abierto llamada MariaDB al ser de código abierto la comunidad mejora el código y el motor de la base de datos y lo más importante baja totalmente los costos de almacenamiento de la data que genera el sistema.

2.4.4.2. Características.

MySQL es un sistema de base de datos relacional, lo que quiere decir que archiva datos en tablas separadas en lugar de guardar todos los datos en un gran archivo, lo que le permite tener mayor velocidad y flexibilidad. Las tablas están relacionadas de

formas definidas, por lo que se hace posible combinar distintos datos en varias tablas y conectarlos.

Todo tipo de persona puede utilizar MySQL y crear modificaciones bajando el código fuente con el fin de ajustarlo a su medida y para su uso, ya que es un software de código abierto. Se establece el GPL (General Public License) en donde se define lo que un usuario puede realizar o no con el software. Si un usuario no se ajusta a las medidas del GPL o requiere el uso del software para aplicaciones comerciales, se puede comprar una versión comercial licenciada.

Inicialmente MySQL carecía de características que podrían considerarse esenciales de las bases de datos relacionales, como integridad referencial y transacciones, aunque a pesar de esto los desarrolladores web comenzaron a utilizarla debido a su simplicidad. A lo largo de las versiones y actualizaciones, estos requerimientos se han ido incorporando al software.

2.5. Servicios REST.

2.5.1. Introducción.

En la actualidad la nueva generación de aplicaciones WEB comparadas con las anteriores son apodadas como la WEB 2.0, ya que estas poseen la ventaja de los Servicios REST y hacen que el ciberespacio sea muy diverso, lo cual se puede observar en el gran número de aplicaciones afianzadas en la nube. Los servicios WEB permiten que los sistemas sean heterogéneos y se comuniquen entre sí utilizando mensajes, lo que genera la posibilidad de que éstos manejen su propio lenguaje para comunicarse, es decir que existe una interoperabilidad entre ellos. Para dar formato a estos mensajes se usa XML debido a que es el lenguaje estándar entre aplicaciones y subsistemas al momento de utilizar los servicios para aplicaciones web. REST se ha vuelto más popular entre los desarrolladores de aplicaciones WEB a causa de su simplicidad, y el uso extensivo de tecnologías basadas en la web como HTTP que es el protocolo más popular y utilizado en el mundo.

2.5.2. Ventajas y desventajas.

En el proyecto actual usaremos REST siendo el recurso más usado para la comunicación de aplicaciones heterogéneas comparando con SOAP son iguales sabiendo que pueden estar en diferentes plataformas cumpliendo el mismo objetivo que es la comunicación de sistemas mediante mensajes.

Como principal mensaje REST permite innumerables tipos de datos dando como opción al desarrollador que use JSON que comúnmente es mucho más rápido y da el mejor soporte al cliente.

Al contrario, SOAP permite la comunicación, pero estructurada esto puede ser malo en caso específicos que requieren data que no siempre está disponible, además SOAP solamente acepta XML y tiene 3 secciones.

- Sobre (Envelope): Define que hay un mensaje y que hay que hacer para procesarlo
- Reglas de codificación: Expresa las instancias de tipos de datos.
- Convención: Representa llamadas a procedimientos y respuestas.

La principal desventaja de REST es la seguridad puesto que no está fuertemente tipado y como método de seguridad solo se puede usar los diferentes verbos que tiene para poder acceder.

El estándar en mensajería REST puede ser una ventaja, pero en consumo de servicios externos puede ser una desventaja porque cada sistema puede definir su estructura y sus tipos de datos, lo cual en sí no es un problema, pero como usualmente se utilizan otros lenguajes, se debe serializar a REST y esto genera pérdidas de tiempo, que se podrían emplear en mejorar un sistema.

2.6. Modelo vista controlador (MVC)

2.6.1. Introducción

MVC es un patrón de arquitectura de software que separa el acceso a datos de la lógica del negocio de un software compuesto de tres componentes Modelo, Vista y Controlador se basa en la reutilización de código y separación de conceptos lo que buscan facilitar el mantenimiento y que el software sea robusto.

- Modelo: es la representación de la información es decir donde se almacena y en que forma se almacena los datos.
- Controlador son los eventos que disparan un conjunto de funciones que ejecuten una acción o guarden datos. Aquí está la lógica de negocio del software con sus reglas, restricciones y algoritmos para procesador la data.

- Es una interfaz de usuario para interactuar con las diferentes funcionalidades que tenga el software.

2.6.2. Uso en la WEB

MVC históricamente tiene orígenes en aplicaciones stand-alone con el avance de las tecnologías de la información este patrón ha ido innovando, siendo adaptado a la WEB, hoy en día es muy utilizado para el desarrollo de frameworks que facilitan el desarrollo de software popularizando el patrón MVC inculcándose en la comunidad de desarrollo de software como estándar internacional.

Capítulo 3 Selección de Herramientas.

En este capítulo se explica detalladamente porque se elige cada tecnología para desarrollan en comparación con otras las ventajas que tienen y lo innovadoras que son en la actualidad también se toma en cuenta los costos de desarrollo y de alojamiento que se debe tomar en cuenta para tener en producción.

3.1.¿Porque PHP?

PHP fue creado para realizar un seguimiento de visitantes en las páginas personales, pero actualmente es un referente en lenguaje script, siendo el más utilizado por usuarios de Open Source, además existe una gran variedad de aplicaciones basadas en PHP que hace que su uso sea sencillo, es decir que no se necesite un conocimiento extenso de esta herramienta para poder usarla. Otra de las ventajas es que se puede crear aplicaciones a partir de guiones o scripts que no necesitan ser compilados y pueden ser insertados directamente en el código de HTML, y su licencia permite un uso libre con muy pocas restricciones.

3.2.¿Porque servicios en REST?

A pesar de que SOAP con el pasar del tiempo se ha posicionado bien alto siendo la tecnología preferida por muchas empresas, para otras es muy compleja y poco flexible, siendo esta razón, se ha empezado a usar REST en cantidades de datos masivas donde traducir SOAP demandaba recursos y tiempos de respuesta al cliente, ambas arquitecturas de la información tienen sus nichos definidos no obstante REST será quien tendrá mucha más aceptación.

En el proyecto actual se ha definido REST por las cantidades de información y la compatibilidad con el consumo de servicios que abastecen al sistema teniendo un sistema integral adaptable a cambios y pensado para que se comunique con otros sistemas en el futuro.

Los servicios REST necesitan escalar para poder satisfacer una demanda en constante crecimiento. Se usan clusters de servidores con balanceadores de carga y alta disponibilidad, de tal manera para crear una topología de servidor, que permita transferir peticiones de un equipo a otro para disminuir el tiempo total de respuesta de una invocación al servicio web. El uso de servidores intermedios para mejorar la escalabilidad hace necesario que los clientes de servicios web REST envíen peticiones completas e independientes; es decir, se deben enviar peticiones que incluyan todos los datos necesarios para cumplir el pedido, de manera que los

componentes en los servidores intermedios puedan redireccionar y gestionar la carga sin mantener el estado localmente entre las peticiones.

Una petición completa e independiente hace que el servidor no tenga que recuperar ninguna información de contexto o estado al procesar la petición. Una aplicación o cliente de servicio web REST debe incluir dentro del encabezado y del cuerpo HTTP de la petición todos los parámetros, contexto y datos que necesita el servidor para generar la respuesta. De esta manera, el no mantener estado, mejora el rendimiento de los servicios web y simplifica el diseño e implementación de los componentes del servidor, ya que la ausencia de estado en el servidor elimina la necesidad de sincronizar los datos de la sesión con una aplicación externa.

No siempre REST es la mejor opción. Está surgiendo como una alternativa para diseñar servicios web con menos dependencia en middleware propietario (por ejemplo, un servidor de aplicaciones), que su contraparte SOAP y los servicios basados en WSDL. De algún modo, REST es la vuelta a la Web antes de la aparición de los grandes servidores de aplicaciones, ya que hace énfasis en los primeros estándares de Internet, URI y HTTP. Como ya analizamos XML sobre HTTP es una herramienta muy poderosa que permite la comunicación de aplicaciones internas, con interfaces basadas en JavaScript Asíncrono y XML (AJAX) puedan conectarse, ubicar y consumir recursos. De hecho, es justamente esta gran combinación con AJAX que generó esta gran atención que tiene REST hoy en día.

Resulta muy flexible el poder exponer los recursos del sistema con un API REST, de manera de brindar datos a distintas aplicaciones, formateados en distintas maneras. REST ayuda a cumplir con los requerimientos de integración que son críticos para construir sistemas en donde los datos tienen que poder combinarse fácilmente y extenderse. Desde este punto de vista, los servicios REST se convierten en algo mucho más grande.

3.3. ¿Porque base de datos relacional MySQL?

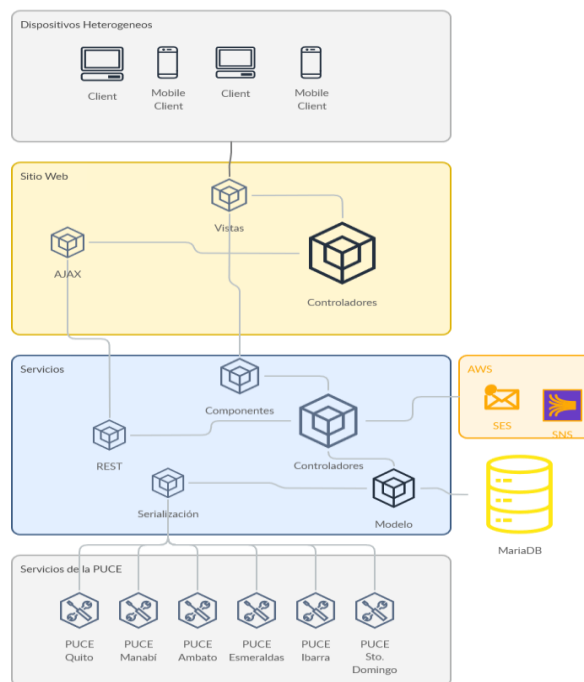
Al momento de elegir en que tecnología vamos a desarrollar un software el punto más crítico es elegir el motor de base de datos del sistema y comúnmente se toman decisiones erradas en base a esta decisión nacen más interrogantes como cual es más fácil de administrar, o cual es suficiente para la cantidad de usuarios que manejará el sistema también en costos cual de la infinidad de motores de bases de datos es el más económico y que abarque todos los recursos que necesite el sistema.

Planteando todas estas variables la posibilidad de MySQL es la óptima porque los costos de licenciamiento de MySQL son nulos solo el costo para el proyecto al usar MariaDB los costos son solo de almacenamiento y procesamiento en el servidor de producción al ser una base de datos relacional consolida la información con claves primarias e indexación además PHP viene por defecto con las extensiones y complementos necesarios para usar MySQL como base de datos para el sistema.

3.4. Arquitectura.

La arquitectura se encuentra conformada de varias tecnologías cuya finalidad es proporcionar una zona en la que distintas aplicaciones se comuniquen entre sí (como se observa en la Ilustración 1), esto es muy útil al momento de generar los datos necesarios tanto para la PUCE como para el administrador de sistemas, pero aun así los servicios que se generan mediante este proceso no son suficientes para notificar y comunicar los eventos existentes, debiendo adquirir dichos servicios de otros proveedores como mail de Amazon y notificaciones push SES y SNS.

Ilustración 1: Diagrama de arquitectura del sistema.



(Robalino, 2019)

Ilustración 1. En el cuadro se pueden observar las distintas tecnologías que hacen del sistema un espacio en el que puedan existir aplicaciones que se comuniquen entre sí.

3.5. Ciclo de vida.

3.5.1. Fase de Exploración.

Esta fase busca saber cuáles son las necesidades que tiene el cliente como usuario, ya que en éstas se basará el programador para realizar el primer prototipo del sistema. Además, todos aquellos involucrados en el desarrollo, analizarán y estudiarán las tecnologías y herramientas que se necesitan para la elaboración del sistema. En esta fase se entrega el artefacto denominado SRS documento de requerimientos

3.5.2. Fase de Planificación en la entrega.

En esta fase el cliente es quien determinara las prioridades, es decir las partes del sistema que los usuarios necesitan con mayor urgencia, teniendo en cuenta el equipo del que dispone, el tiempo que se empleará y el esfuerzo estimado para la elaboración. Una vez que el cliente establece las prioridades, se puede llegar a un acuerdo con los programadores para el desarrollo de la primera versión, cuya codificación no se debe tardar más de tres meses, eso sí tomando en cuenta la dificultad y el tamaño del proyecto a realizar. En esta fase se entrega el diagrama de Gantt

3.5.3. Fase de desarrollo de Iteraciones.

En cada proyecto a realizar existen un número variado de iteraciones que son pequeños procesos que se llevan a cabo para llegar a la entrega final. Por lo general las iteraciones no deberían durar más de tres semanas, y la primera semana sirve para instaurar la forma en que se elaborará el sistema o programa, que servirá de ejemplo para la programación del software en cuestión, todo esto obviamente teniendo en cuenta las demandas del cliente, quien en una instancia final será el que determine las ejecuciones que se desarrollarán en cada una de las iteraciones, cuando la última iteración esté lista y aprobada, el sistema entrará en fase de producción. Para el inicio de cada iteración debemos fijarnos en algunos aspectos como son, los requisitos del usuario no resaltados, tareas sin terminar y bugs⁷ en las iteraciones que se realizaron con anterioridad, además se deberá recalcular el tiempo de elaboración del proyecto. Se entregará código fuente y diagramas de clases de cada iteración. En esta fase se entrega la codificación

⁷ BUG: Es un defecto en el software, que hace que un programa funcione incorrectamente.

3.5.4. Fase de producción.

Previamente a generar un producto o programa final, que esté alineado a los requerimientos marcados por el cliente, se deberán dar pruebas de los resultados obtenidos, además la posibilidad de agregar complementos que sean requeridos y que respondan a los cambios que suceden en las iteraciones previas, con el fin de generar un producto final de calidad que apruebe el cliente.

3.5.5. Fase de mantenimiento.

Posterior a una versión preliminar del ciclo de vida se deberá incrementar iteración dependiendo de las necesidades del cliente. En esta nueva fase, se podrá ampliar las funciones o los requerimientos del proyecto para que así soporte futuros cambios en el proyecto haciendo un producto de alta disponibilidad.

3.5.6. Fase de muerte del Proyecto.

Esta constituye la fase final, en la cual el desarrollo del producto se ha consumado, es decir que el consumidor no tiene más correcciones o implementaciones para el mismo, por lo tanto, no necesitamos realizar más cambios en su distribución. También puede morir porque no se pueden crear más beneficios, o porque no se dispone de los recursos financieros necesarios para ejecutar más cambios. En esta fase se entrega el manual de usuario.

3.6. Costos.

Los costos de alojamiento de un servidor estándar pueden ser muy variados (como se observa en la Tabla 1, sin contar con los servicios adicionales que necesita el sistema, por ejemplo, un servicio de notificaciones, como el de envío de correos.

Solo un servidor LEMP (Linux Nginx MySQL PHP) es suficiente, ya que resulta ser el más ágil y eficiente para este desarrollo, el punto decisivo para la elección de este servidor, fue el costo de mantenimiento que es de 60 dólares anuales por un servidor básico para la primera versión del sistema, otro punto de criterio un poco sesgado es que, la dirección de Informática de la PUCE pone como requisito que el sistema debe tener como base de datos MySQL y este tipo de servidor se acopla perfectamente a ese requerimiento y a las necesidades actuales del sistema.

Tabla 1: Costos de servidores

	LEMP	Microsoft IIS	Tomcat	Ruby	Python
Alojamiento	\$20	\$150	\$132	\$36	\$60
Base de datos		\$10			
Dominio	\$20	\$20	\$20	\$20	\$20
SSL	\$20	\$20	\$20	\$20	\$20

(Robalino, 2019)

Tabla 1. Precios de alojamiento, base de datos, dominio y SSL de distintos servidores.

3.7.Herramientas utilizadas.

Herramienta	Tipo	Versión
XP	Metodología	-
PHP	Lenguaje de programación	7,0
MySQL	Motor de Base de datos	5,0
PowerDesigner	Modelador de Datos	16,6
PhpStorm	IDE	
Cloud Linux	SO	7,2

(Robalino, 2019)

Tabla 2. Herramientas

Capítulo 4 Sistema de Bolsa de Empleo y Pasantías PUCE.

Este capítulo trata sobre el desarrollo del sistema de Bolsa de Empleo, es decir vamos a hablar sobre los documentos que se han generado por cada iteración, además dialogaremos de los elementos que contiene cada iteración de XP, y finalmente discutiremos las funcionalidades que se han generado por fase

4.1. Administración de Roles.

4.1.1. Estudiante.

El estudiante es el actor protagonista del sistema porque a través de su información (Hoja de vida) va a nutrir al sistema, permitiendo que se obtengan datos, estadísticas para el Consejo de Aseguramiento de la Calidad de la Educación Superior (CACES), nos ayudará también a determinar quién usa la plataforma y quien consiguió trabajo a través de este medio, lo cual contribuye a fortalecer el sistema, porque éste se verá dotado de un sinnúmero de perfiles en los que además se podrá encontrar habilidades muy variadas para las empresas registradas, permitiéndoles elegir a la persona que mejor se acople a sus necesidades, que puede ser un estudiante de la PUCE o un ex estudiante. Por ahora están divididos en: estudiantes que tiene menos del 99% de aprobación de la malla curricular de cada carrera, los egresados que son los que tiene completa la malla curricular pero no tiene trabajo de titulación y por ende no tiene título profesional, y por último los graduados que tiene la malla aprobada y el trabajo de titulación final de la carrera que son los profesionales.

4.1.2. Coordinador.

Coordinadores son todos los docentes que forman parte del grupo de tutores de prácticas preprofesionales, son ellos los que a través del sistema pueden revisar y socializar publicaciones de cada carrera a aquellos estudiantes que tengan a cargo, es decir los que estén realizando sus prácticas con dicho tutor, en el sistema el coordinador solo puede ver las publicaciones, el listado de estudiantes registrados en su facultad, y las estadísticas que necesitan para reportar al CACES sobre empleabilidad de cada carrera. Esto es posible debido a que el sistema cuenta con requerimientos como prácticas preprofesionales y pasantías.

4.1.3.Reclutador de la Empresa.

Es la persona de cada empresa quien registró a su organización en el sistema quien por lo general es una persona de recursos humanos, un especialista en selección que se encarga de ingresar la información de su organización para que el administrador la verifique y valide para mayor seguridad de cada estudiante, es decir evitar organizaciones duplicadas o fantasmas. El reclutador es el encargado en subir su requerimiento al sistema, y publicarlo en el universo que se ajuste al perfil duro que requiera, luego de subir la publicación, los estudiantes aplicarán y el reclutador elegirá un estudiante del conjunto de personas que respondieron a dicha difusión.

4.1.4.Administrador del Sistema.

El administrador del sistema estará atado a soporte, será el encargado de solucionar problemas comunes como dar de baja a una publicación o validar la información de una empresa, también estará a cargo de activar el día de feria de empleos donde las empresas podrán tener un repositorio de hojas de vida, solo con el número de cédula de cada estudiante.

4.2.Ciclos de vida

Los ciclos de vida están planeados en base a la arquitectura del sistema y las iteraciones que se realizan en corto tiempo, las cuales están sujetas a revisiones de los actores que intervienen en el sistema, pero tomando en cuenta que no todos los actores de producción pueden intervenir en todas las partes que componen el sistema, sino que los distintos tipos de actores cumplen un determinado rol.

4.2.1. Fase de Exploración.

En esta fase se busca la forma de obtener y unificar la data que existe actualmente en la PUCE, en orden de poseer la información necesaria de cada estudiante, pero que además esté validada por la PUCE. Para esto, cada sede debe dotar de una base de datos de sus estudiantes, en donde cada alumno perteneciente a dicha sede se pueda registrar. Para aquellas sedes que en el peor de los casos no dispongan de un api REST o SOAP y que dispongan de la información masiva de estudiantes, se creó un servicio genérico que ayuda a solventar este problema, permitiendo que la información solicitada pueda ser entregada en cualquier otro servicio web, pero con la condición de que dicha sede debe dotar de un manual de uso del servicio web usado,

cuyo contenido sean los datos que estén destinados para la Bolsa de Empleos. El resultado de esta fase se presenta en el [Anexo 1](#).

4.2.2.Fase de Planificación de la Entrega.

Existen tres parámetros diferentes a tomar en cuenta para la planificación de la entrega, que son: la prioridad que se le haya asignado a la funcionalidad, en donde las de prioridad alta serán las que tengan más importancia dentro de este grupo, las necesidades y el tipo de reporte, siendo las dos últimas valoradas de acuerdo con las necesidades del coordinador de cada unidad académica, quien posteriormente realizará la validación del reporte de los avances, esto se logra a través de las diferentes versiones del proyecto que se subirán a GitHub para que el coordinador pueda apreciar los avances realizados, y de esta manera apruebe los reportes en tiempo real, lo que resulta ideal para acortar los tiempos de entrega del sistema dentro de la planificación del tiempo de desarrollo. El resultado de esta fase está en el [Anexo 2](#)

4.2.3.Iteraciones.

Para el plan de iteraciones se debe tomar en cuenta, que éstas o las tareas que la componen no estén inconsistentes, sin aprobación o con errores, las iteraciones están planeadas en un lapso de 5 días hábiles, cuando la última iteración sea aprobada, el sistema estará listo para producción, es decir que contará con la preaprobación del coordinador de Bolsa de Empleo y que las historias de usuario no tendrán errores.

4.2.3.1.Hoja de Vida de los Estudiantes.

La hoja de vida de cada estudiante de la PUCE es la materia prima para el sistema de Bolsa de Empleo, ya que con la información de cada estudiante el sistema se nutre de la data necesaria para poblar la base de datos. Una vez registrados los currículos de cada alumno, las empresas eligen a sus aspirantes, teniendo en cuenta que deben cumplir el perfil mínimo de aplicación de los requerimientos, y de esta manera contratarlos en su organización.

Además, estos antecedentes constituyen un indicador para cada coordinador de Prácticas Preprofesionales, ya que les permite saber quiénes completaron sus prácticas y a quienes les falta este requerimiento para graduarse. En el [Anexo 3](#) se presenta el diagrama de clases asociado con esta funcionalidad.

4.2.3.2. *Publicaciones de Empresas.*

Las publicaciones o requerimientos de las Empresas son el aperitivo para incentivar a los estudiantes a llenar su hoja de vida. Cada requerimiento que se registre en el sistema será necesariamente etiquetado dependiendo del tipo de requerimiento que es y además especificará el nivel de instrucción que los aplicantes necesitan.

Una publicación puede ser divulgada en varias unidades académicas, ya que una oferta puede ser afín con distintas carreras. En el Anexo 4 se presenta el diagrama de clases asociado con esta iteración.

4.2.3.3. *Aplicación a Publicaciones.*

La aplicación de estudiantes a las publicaciones es la parte fundamental del sistema, ya que a partir de este punto se genera todo el proceso de información, flujo de seguimiento y contratación de un estudiante.

Existen dos estados de aplicación, la primera es la invitación, que es cuando una organización invita a un estudiante a aplicar a su publicación, y la segunda es la aplicación de un estudiante que es cuando éste aplica a la oferta de una organización. Cuando un estudiante aplica a una publicación envía su hoja de vida completa con la información de contacto a la organización, en cambio cuando el alumno es contactado por una invitación la organización que lo solicitó únicamente puede ver la información básica de la hoja de vida. En el Anexo 5 se presenta el diagrama de clases asociado a esta iteración.

4.2.3.4. *Reporte.*

Todos los procesos que hace el sistema se resume en este reporte, el cual tiene dos versiones: un reporte global en tiempo real, del número de estudiantes registrados en el sistema con aplicaciones por unidad académica y otro reporte semestral que genera el sistema para cada coordinador con información de cada unidad académica en el periodo de aplicación del semestre. En el Anexo 6 se presenta el diagrama de clases asociado a esta iteración.

4.2.4. Producción.

En la fase de producción está establecido hacer pruebas de carga y consumo masivos de los servicios web que genera el sistema, posteriormente se realizarán pruebas con conexión

limitada a internet, con la finalidad de que se puedan optimizar peticiones web en Ferias de Empleo que tengan una conexión a internet muy mala, ya que de esta manera el rendimiento del sistema será bueno independientemente de la calidad de internet. En el anexo 7 se presenta el documento de primera salida a producción

4.2.4.1. *Back-End.*

MySQL permite un sistema con alta disponibilidad lo que hay que tener en cuenta es que hay que configurar todas las opciones que como motor de base de datos tiene disponibles, también un factor importante al momento de hacer una conexión al motor de base de datos con una configuración de conexión óptima considerando el tipo de conexión, firma de seguridad y lo más importante es la codificación de caracteres por los diferentes tipos que existen en el lenguaje natural con sus símbolos por cada idioma, también cerrar la cadena de conexión con el motor de base de datos para que no genere tráfico innecesario en el motor de base de datos.

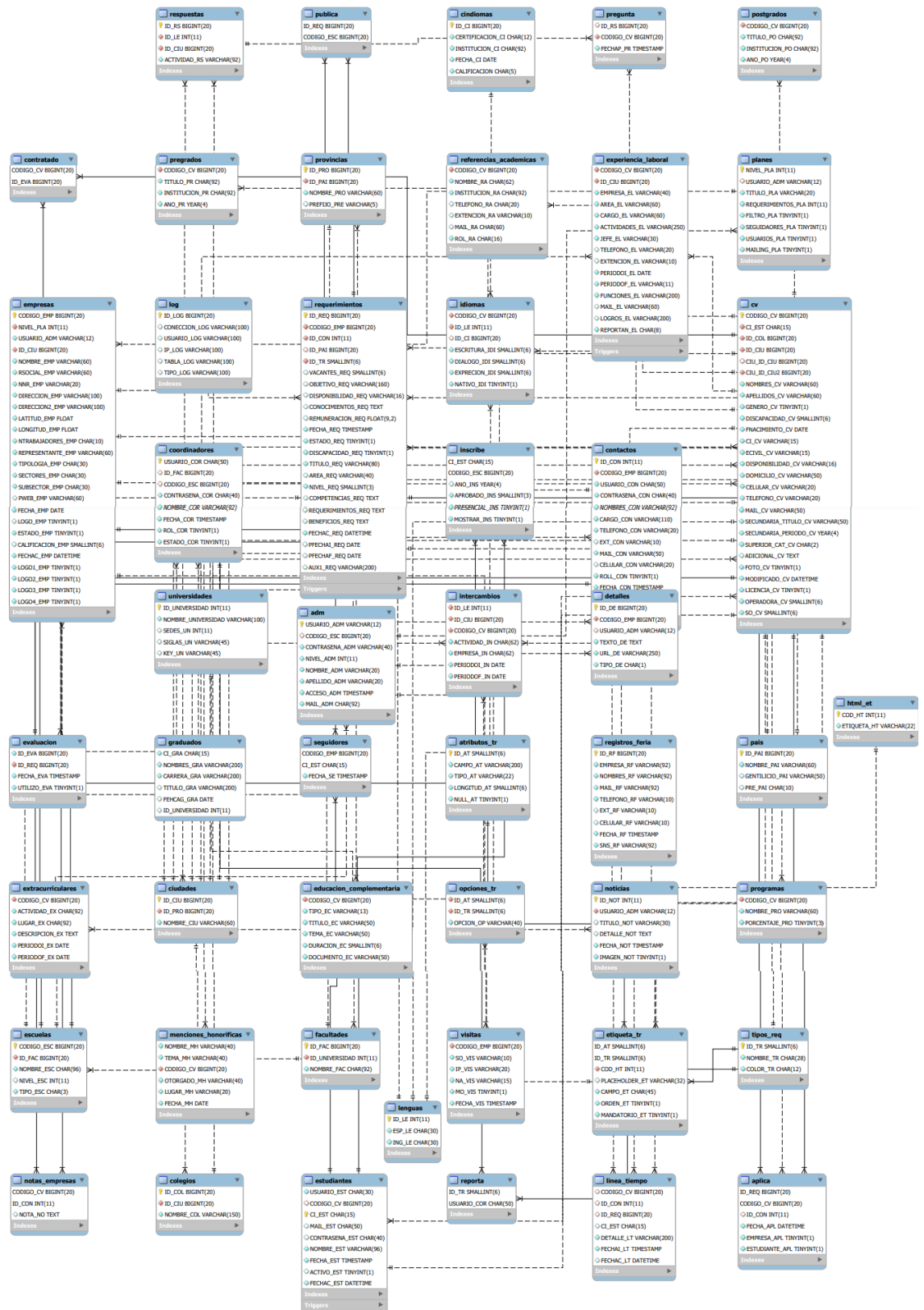


Ilustración 2 Modelo entidad relaciones

(Robalino, 2019)

4.2.4.2. *Front-End.*

El conjunto diverso de tecnologías usadas en este proyecto converge en una sola para interpretar la información que genera el sistema, de forma que el usuario pueda consumir dicha información. Como ya es bien sabido al usar HTML5 hablamos del uso de JavaScript como lenguaje de programación, razón por la cual, se vuelve necesario el uso de librerías para el manejo del DOM, ya que de esta manera no usaremos el lenguaje nativo obligatoriamente, sino que haremos que la programación sea mucho más sencilla de lograr en otro lenguaje que es JQuery.

Otro punto que destacar es que está planeado que el sistema funcione en dispositivos con diferentes dimensiones de pantalla, es decir que el sistema sea multiplataforma siempre y cuando el cliente acceda desde un navegador, pero para que estas herramientas conmuten entre si y puedan convivir entre ellas se requiere el uso de JQuery conjuntamente con un framework CCS que se utiliza para diseño de pantallas multiplataforma Bootstrap y un conjunto de librerías extra que necesita el sistema para funcionar, logrando así que el usuario pueda ingresar la información necesaria, con los componentes necesarios para que interactúen con él.

4.2.5. Mantenimiento.

El sistema cuenta con catálogos de información genérica como Nacionalidad, Idioma, etc. En el caso que un usuario necesite añadir un registro a estos campos se planifica usar un soporte en línea que ayude a solventar y validar la información que se ingresen en estos catálogos, para tener una base de datos consolidada. Se considera que con el tiempo el nivel de soporte baje exponencialmente, ya que se supone que por cada incidente se solucionan futuras posibles incidencias. El cliente verificó y validó las funcionalidades requeridas implementadas en el sistema, dio su aceptación para poner en ambiente de producción el sistema. En esta fase no se generó artefacto.

4.2.6. Muerte del Proyecto.

Está contemplado que el sistema después de la fase de producción no va a necesitar mantenimiento, aparte de los servicios que consume, que si necesitarán mantenimiento, como por ejemplo SES y SNS que requieren de pagos mensuales, y validación de endpoints. Ver Anexo

Una vez finalizada la fase de producción, el sistema va a necesitar mantenimiento esporádico, a diferencia de los servicios como SES y SNS que requieren de pagos mensuales, y validación de endpoints. En el anexo 8 que se encuentra en el CD, está el manual de usuario

Capítulo 5 Conclusiones y Recomendaciones.

5.1. Conclusiones

- Al desarrollar sistemas multicapa hay que considerar varios aspectos desde todas las perspectivas posibles, más aun si están enfocados para un entorno web, considerando en primer lugar, quienes y en que dispositivo van a usar el sistema, también el tipo de conexión a internet con la que contarán los clientes, para poder decidir según estas variables cuáles son las herramientas que pueden usar los clientes, y que además puedan convivir en el servidor de aplicaciones que se decida implementar para el sistema.
- Los servicios dotados por la PUCE para el sistema de bolsa de empleo son muy buenos, sin embargo, cuando existe una carga masiva de información como sucede con el servicio SOAP de la PUCE sede Ambato, se ralentiza el servidor de aplicaciones por lo que se optó por hacer una tarea programada en las noches y que no afecte el procesamiento de los sistemas alojados en el servidor.
- Al desarrollar software para una institución hay que tener en cuenta los estándares y la guía de imagen que tiene la misma para cada pantalla, además de la identificación del sistema en el Internet.
- Si bien las herramientas nos ayudan a desarrollar software de calidad hay q tener en cuenta que este esta susceptible a cambios que quiera realizar el cliente, y por tanto se debe crear ambientes de pruebas y de producción, de esta manera tanto el desarrollador como el cliente se verán beneficiados cada uno a su manera, debido a que el consumidor podrá palpar los avances del proyecto para aprobarlos o realizar cambios en la estructura del software de manera eficaz y oportuna, a través de la comunicación con el desarrollador.
- La metodología y los ciclos de vida se adaptaron perfectamente al proyecto porque se logró involucraron a todas las personas que realizaban este flujo de forma manual y ahora lo pueden hacer a través de la plataforma haciendo de esta metodología Ideal para la participación de los clientes en el desarrollo de Software.

- La plataforma para la primera versión cumple todo los aspectos y requerimientos de la primera fase para una segunda versión se estima tener recomendaciones e interacción entre estudiantes.

5.2.Recomendaciones

- Se recomienda que por cada cambio sugerido por el cliente se versione el código bajo Git, que el cambio del cliente se vea reflejado en un commit con detalles y si es posible que conste de firma electrónica del cliente para evitar problemas futuros.
- El uso de Latex en el proyecto es una opción excelente ya que, al generar las hojas de vida en PDF, proporciona un valor agregado al sistema, por lo tanto se recomienda que para posteriores versiones se mantenga el uso estándar de Latex.
- El uso de SQL es indispensable, pero para proyectos donde hay varias tablas, se debe tener en cuenta el uso de un ORM, para el manejo de información de la base de datos.
- En el desarrollo de este proyecto en busca de un entorno de producción potente, personalizado y a la vez auto mantenible se pasó desde GoDaddy a Dreamhost buscando un servidor que no necesite mantenimiento de Sistema Operativo lo cual se encontró en estos servidores, pero no contaba con herramientas adicionales a las que necesitaba el proyecto. Después se optó pasar todo el alojamiento a Amazon fue una ventaja al tener un entorno totalmente personalizado el cual necesitaba un administrador, con el tiempo esto fue un problema porque necesitaba parches de actualización de motor de base de datos y de sistema operativo para el proyecto esto era desgastante. Por último, se optó por pasar todo el alojamiento de base de datos y de servidor de aplicaciones a WebFaction el cual resultó ser el más eficiente porque era personalizable a nivel que se puede instalar cualquier cosa que necesite el sistema y no es necesario que un administrador de mantenimiento a un servidor porque este servicio ya incluye Webfaction además de eso soporta todos los lenguajes existentes en la web.

ANEXOS.

Anexo 1.

Documento de Requerimientos

Introducción

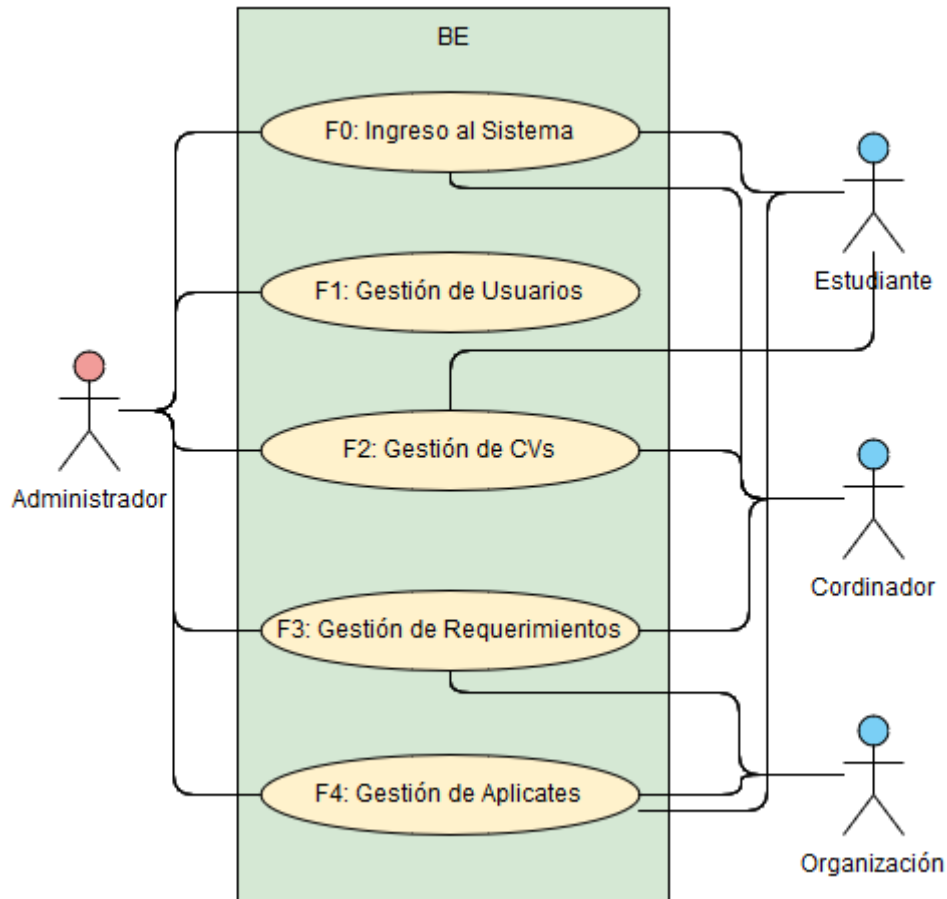
Actualmente, la forma de buscar empleo ha evolucionado, se ha vuelto casi nula la búsqueda de trabajo en medios impresos, y la mayoría apuesta por publicar y buscar ofertas laborales en la web. La bolsa de empleo de la PUCE está diseñada para facilitar esta búsqueda tanto a estudiantes como a egresados de esta Universidad, y lo hace mediante el manejo de su hoja de vida, la cual compara con los requerimientos solicitados por las diferentes empresas que se encuentren ofertando empleos en ese momento, y de esta manera poder ofrecer la opción de postular a esta publicación, únicamente a aquellos estudiantes que cumplan con el perfil, es decir en otras palabras filtrar las ofertas, de forma que los estudiantes o profesionales puedan ver únicamente las que les interesa. Como existen varias empresas afiliadas a este sistema, aquellos que forman parte del sistema de bolsa de empleo de la PUCE se enteran de las ofertas antes que otros profesionales o estudiantes que no pueden acceder a esta, y por tanto a la hora de encontrar trabajo les aporta una gran ventaja.

Este sistema además de proporcionar opciones tanto a las empresas como a sus estudiantes y egresados, es capaz de generar informes de empleabilidad que reflejan el porcentaje exacto de los estudiantes que han sido contratados en cada unidad académica, la cual es información que se reporta al sistema nacional de información de educación superior del Ecuador.

Glosario

- PUCE: Pontificia Universidad Católica del Ecuador

Requerimientos Funcionales



Historias de Usuario

Historia de Usuario			
Número	1	Nombre	Gestión de Usuarios
Usuario	Administrador	Iteración Asignada	1
Prioridad	Alta	Riesgo	Bajo
Descripción			
El usuario que tiene ingreso al sistema sea estudiantes coordinadores o administradores debe contar con un doble respaldo en el caso de que al registrarse hayan puesto un correo electrónico no valido y tengan q recuperar su cuenta de alguna manera, y esta funcionalidad esta destinada para que un administrador pueda cambiar esta información que solo el sistema genera y solo este actor puede manipular.			
Observaciones			
El modulo de gestión de usuario estará habilitado solo para ciertos administradores y solo será para que puedan manipular la información de los usuarios mas no podrán ver claves de acceso.			

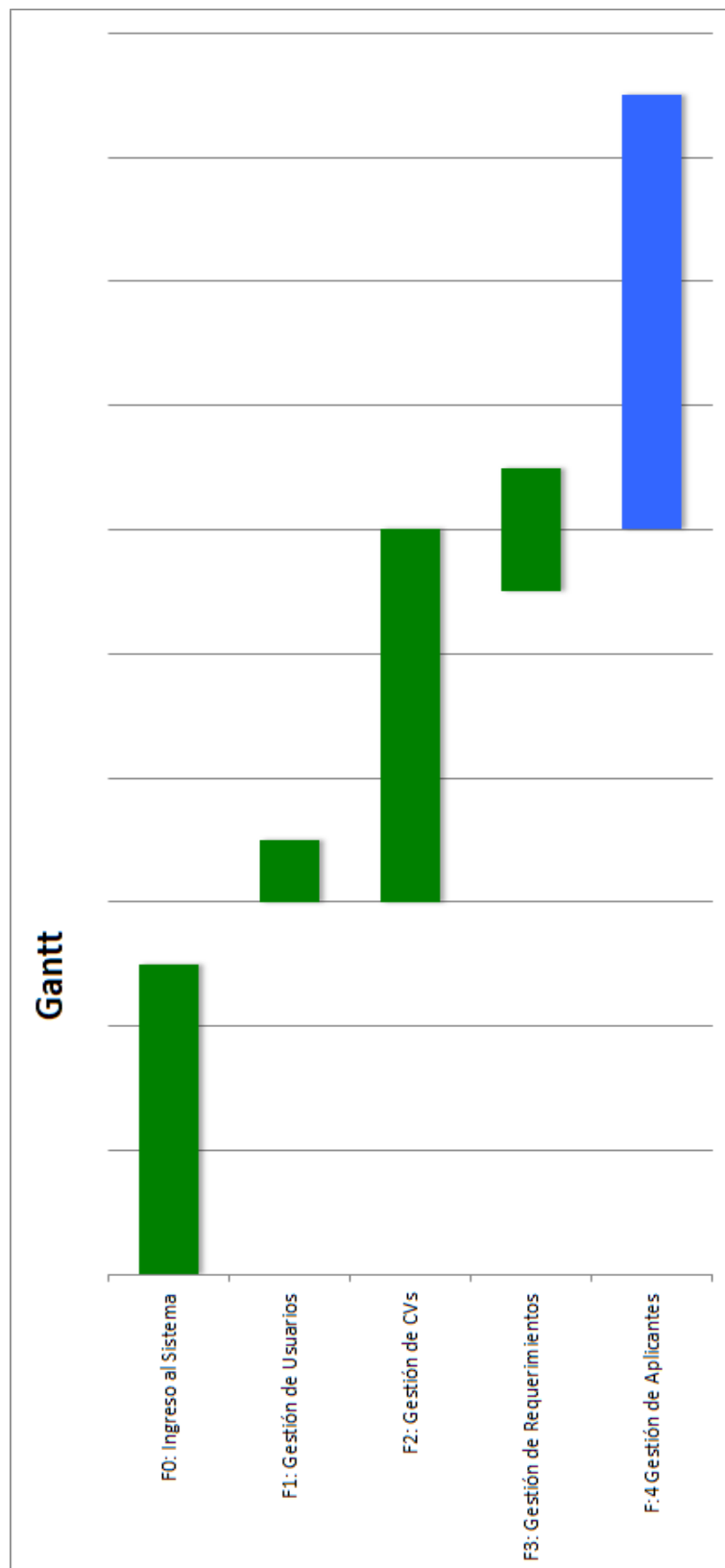
Historia de Usuario			
Número	2	Nombre	Gestión de CVs
Usuario	Estudiante	Iteración Asignada	1
Prioridad	Alta	Riesgo	Alto
Descripción			
El estudiante debe registrar su información en el sistema para que las empresas puedan ver y evaluar el perfil de cada estudiante.			
Observaciones			
El modulo de CVs esta habilitado para que solo los estudiantes puedan ingresar datos los demás actores solo pueden ver esta información			

Historia de Usuario			
Número	3	Nombre	Gestión de Requerimientos
Usuario	Empresas	Iteración Asignada	2
Prioridad	Alta	Riesgo	Alto
Descripción			
Las Organizaciones registradas debe ingresar las publicaciones con el perfil duro de la vacante disponible y así los estudiantes puedan aplicar a la oferta laboral.			
Observaciones			
El modulo de Requerimientos esta habilitado para todos mas no todos pueden editar o ingresar información, solo las empresas pueden ingresar información y los estudiantes pueden aplicar, en casos extremos el administrador deberá dar de baja una publicación por motivos extremos.			

Historia de Usuario			
Número	4	Nombre	Gestión de Aplicantes
Usuario	Estudiante	Iteración Asignada	3
Prioridad	Alta	Riesgo	Bajo
Descripción			
El estudiante al momento de aplicar a un requerimiento debe registrar la fecha notificar a la organización y viceversa si la organización invita a aplicar a un estudiante, al finalizar el proceso de selección el sistema deberá preguntar a la organización cuantos estudiantes contrato.			
Observaciones			
Este modulo esta destinado para estudiantes y organizaciones el resto de actores solo pueden ver la información que genera este proceso un Requerimiento dura 60 días calendario.			

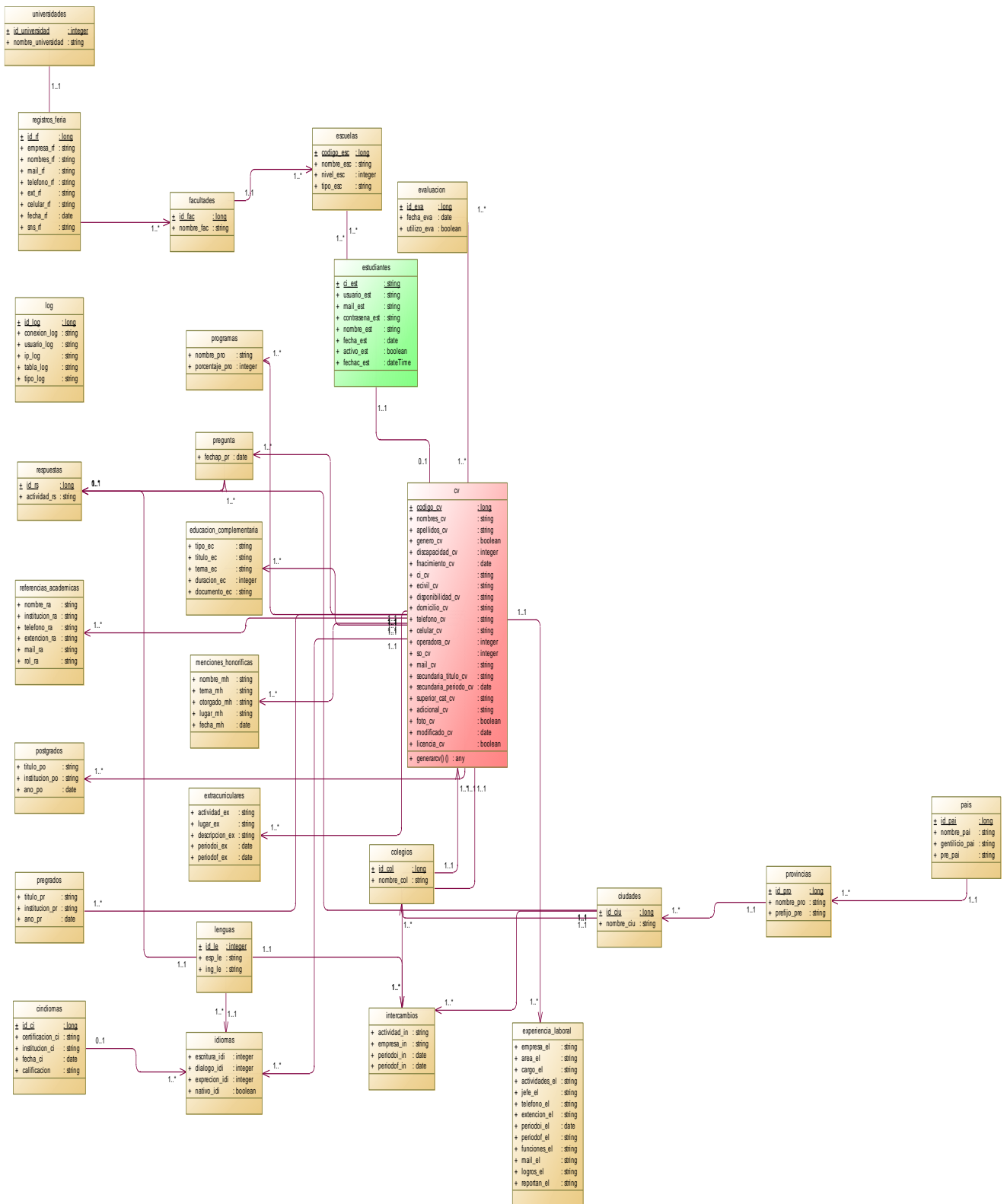
Historia de Usuario			
Número	5	Nombre	Reporte de Aplicantes
Usuario	Coordinador	Iteración Asignada	4
Prioridad	Alta	Riesgo	Alto
Descripción			
Al tener toda la información de todos los procesos se necesita 3 reportes por unidad académica que contenga el numero de ofertas laborales la cantidad de aspirantes y contratados y los aplicantes.			
Observaciones			
Este reporte debe generarse en Excel semestralmente.			

Anexo 2 Diagrama de Gantt



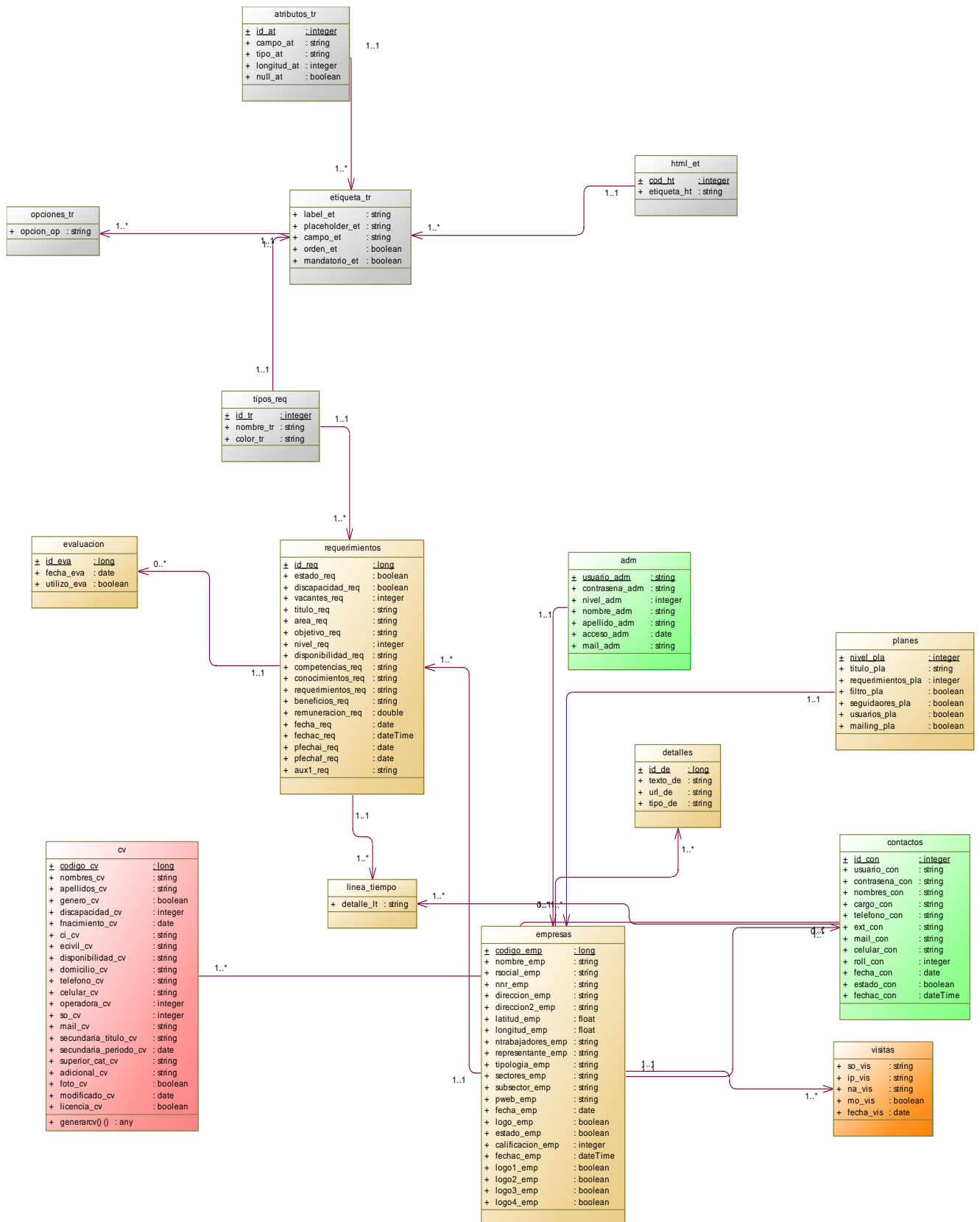
Anexo 3

Diagrama de clases – Hoja de vida de los estudiantes



Anexo 4

Diagrama de clases – Publicaciones de empresas



Anexo 7
Primera salida a producción

servicesuser
E C U A D O R

Ulloa N26-10 y Mosquera Narváez
Tel: (593 2) 510 4056 / 511 5651
E-mail: info@sue.com.ec
www.sue.com.ec
Quito - Ecuador

Quito, 19 de agosto de 2019

A quien corresponda
Presente.-

De mis consideraciones:

Yo, **Manuel Soria** en calidad de **Gerente General** de la empresa **SERVICESUSER Cia. Ltda.** con **RUC 1792140730001**, tengo a bien certificar lo siguiente:

La fecha de inicio de producción **del SISTEMA BOLSA DE EMPLEOS PUCE** fue el 03 de octubre de 2017.

En caso de requerir información adicional, favor remitirse a la información de contacto o directamente a nuestras oficinas.

Atentamente,

SERVICESUSER CIA. LTDA.
RUC 1792140730001

GERENTE

Manuel Soria
Gerente General
SERVICESUSER Cia. Ltda.
0998682227
msoria@sue.com.ec

Bibliografía

- Abeyasinghe, S. (2008). *Restful php web services*. Birmingham: Packt.
- Abrahamsson, P. S. (2002). *Agile software development methods Review and analysis*. Finlandia: VTT Publications.
- Baudrant, A. (2011). *Silicon technologies: Ion implantation and thermal treatment*. Gran Bretaña: ISTE & WILEY.
- Beck, K. (2004). *Extreme Programming explained (12 ed.)*. Boston, San Francisco, Nueva York, Toronto, Montreal, London, Munich, Paris, Madrid, Capetown, Sidney, Tokio, Singapore, Mexico City: Addison-Wesley.
- Cobo, Á., Gómez, P., Pérez, D., & Rocha, R. (2005). *PHP y MYSQL: Tecnologías para el desarrollo de aplicaciones web*. Fernandez - España: Dias de santos.
- Gulabani, S. (2013). *Developing restful web services with jersey 2.0*. Birmingham: Packt.
- Hayward, J. (2011). *Django javascript integration ajax and jquery: Ajax and jquery*. Birmingham: Packt publishing.
- Jeffries, R., Anderson, A., & Hendrickson, C. (2001). *Extreme Programming Installed (3 ed.)*. Canadá: Addison-Wesley.
- Joscowikz, J. (2008). Reglas y Prácticas en Extreme Programming. *Universidad de Vigo*, 22. Obtenido de <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
- Kamthan, P. (2007). *Extreme Programming for Mobile Applications*. Canadá: IGI Global.
- Letelier, P., & Penadés, C. (15 de enero de 2006). Metodologías ágiles para el desarrollo de software: eXtreme programming (XP). *Técnica Administrativa*, 5(26), 13.
- Nedelcu, C. (2010). *NGINX HTTP server*. Birmingham: PACKT.
- Ozar, G. (2012). *Mysql management and administration with navicat*. Birmingham: Packt.
- Peacock, M. (2010). *PHP 5 Social Networking*. UK: Packt PUBLISHING.
- Planificación estratégica PUCE. (1 de Enero de 2011 - 2015). *Pontificia Universidad Católica del Ecuador*. Obtenido de www.puce.com.ec
- Richardson, L., & Ruby, S. (2007). *Restful web services*. Estados Unidos: O'reilly.
- Rico, D. F., Sayani, H. H., & Sone, S. (2009). *The business value of agile software methods maximizing ROI with just-in-time processes and documentation*. USA: J.Ross publishing. Obtenido de https://books.google.com.ec/books?id=anLeaMMgmo0C&printsec=frontcover&hl=es&source=gbs_atb#v=onepage&q&f=false
- Robles, G., & Ferrer, J. (2002). Programación extrema y software libre. En Spalinux (Ed.), *Spalinux* (pág. 23). Madrid: Spalinux. Obtenido de <http://www.mmc.geofisica.unam.mx/Replicas/LuCAS/Presentaciones/200211hispalinux/ferrer/robles-ferrer-ponencia-hispalinux-2002.pdf>
- Sarkar, D. (2011). *NGINX 1 web server implementation cookbook*. Birmingham: PACKT.

Schneller, D., & Schwedt, U. (2010). *Mysql admin cookbook*. Birmingham: Packt .

Tiuquinga Taday, N. (2018). *Análisis de las plataformas OpenSource: NGINX y LIGHTTPD aplicando a la implementación de streaming mediante estándares de calidad en la universidad nacional de chimborazo*. Riobamaba , Chimborazo, Ecuador: UNACH.

Wellman, D. (2013). *Jquery hotshot*. Birmingham: Packt publishing.