



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



PROYECTO DE GRADO

“BOLSA DE TRABAJO CARRERA DE INFORMÁTICA”

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERIA DE SISTEMAS INFORMÁTICOS

POSTULANTE: RICHARD PABLO QUISPE HUIZA
TUTOR METODOLÓGICO: M. Sc. ROSA FLORES MORALES
ASESOR: LIC. VICTORIA HURTADO CERRUTO

La Paz – Bolivia

2012

DEDICATORIA

A Dios por darme las fuerzas y la posibilidad de realizar este Proyecto.

A mis padres Armando y Celestina, por el apoyo en todos los emprendimientos, por inculcarme valores éticos y morales, para ser una persona de bien.

A mi querida esposa, por su inagotable apoyo y su infinita paciencia en todo momento.

A mis hermanas y hermanos, ya que sin el apoyo moral brindado no se hubiera podido culminar satisfactoriamente este trabajo.

AGRADECIMIENTOS

Es fácil decir la palabra “gracias”, pero es difícil expresar realmente la gratitud que se siente, y existen muchas personas a quienes quisiera expresarles mi gratitud, por todo el apoyo brindado, pero con las disculpas del caso y pecando de omisión, quisiera agradecer de manera especial:

A quien me da las fuerzas para vivir y seguir adelante a pesar de las adversidades, quien me llena de energía y positivismo día a día, gracias a Dios por ser mi mayor apoyo.

A la Lic. Rosa Flores Morales por haberme guiado y asesorado durante la elaboración de este proyecto.

A la Lic. Victoria Hurtado Cerruto, quien me tuvo paciencia y comprensión durante la elaboración de este proyecto, gracias por sus consejos y la gran ayuda que me brindó para la culminación del presente proyecto.

A la Jefa de Carrera la Lic. Menfy Morales Rios, quien me tuvo paciencia, comprensión, apoyo, y cuyos consejos fueron muy útiles.

A mis docentes, quienes cumpliendo su labor lograron brindarme los conocimientos necesarios que fueron muy útiles para realizar ciertos contenidos del presente proyecto. Al plantel administrativo de la Carrera de Informática, quienes me ayudaron cumpliendo su labor de la mejor manera.

A mis compañeros de universidad, por brindarme su amistad y apoyo incondicional ya todos estos años de estudio hubieran sido vacíos.

RESUMEN

El presente proyecto de grado fue desarrollado para la Carrera de Informática bajo el título de: “Bolsa de Trabajo Carrera de Informática”, con el objetivo de publicar las ofertas laborales y promocionar a los egresados de la carrera; solucionando así la falta de una herramienta tecnológica que coadyuve a gestionar los requerimientos de empleo, de las instituciones, y de los egresados de la carrera.

La metodología de desarrollo utilizada fue la Programación Extrema (XP), porque se ajusta al desarrollo de este tipo de trabajo, además provee un soporte tecnológico muy robusto. Así también se utilizó la herramienta Enterprise Architect, para generar la documentación que facilitó en gran medida el análisis y diseño del sistema.

Concluido el proyecto, se realizaron las pruebas pertinentes a la etapa de evaluación del software, incluyendo aspectos relacionados con las métricas de calidad propuestas en el estándar ISO 9126. Este resultado demuestra que el producto se encuentra a entera satisfacción del usuario; las pruebas de mantenimiento evidencian que el software podrá ser actualizado de manera oportuna con la facilidad que proveen las herramientas de edición; por otra parte la portabilidad como la usabilidad que el producto podrá ser instalado y utilizado por el personal de la carrera ya que cuenta con la documentación necesaria.

Palabras clave: Bolsa de trabajo, Programación Extrema.

ÍNDICE GENERAL

CAPÍTULO I MARCO REFERENCIAL.....	1
1.1 INTRODUCCIÓN.....	2
1.2 PROBLEMÁTICA.....	3
1.3 OBJETIVOS	3
1.3.1 OBJETIVO GENERAL.....	3
1.3.2 OBJETIVOS SECUNDARIOS	4
1.4 JUSTIFICACIÓN	4
1.4.1 JUSTIFICACIÓN SOCIAL.....	4
1.4.2 JUSTIFICACIÓN ECONÓMICA.....	4
1.5 ESTADO DEL ARTE	5
1.5.1 TRABAJOS SIMILARES.....	5
1.6 METODOLOGÍA.....	6
1.7 TÉCNICAS Y HERRAMIENTAS	6
CAPÍTULO II MARCO TEÓRICO.....	7
2.1 INTRODUCCIÓN.....	8
2.2 BOLSA DE TRABAJO	8
2.3 INGENIERIA DE SOFTWARE	9
2.4 METODOLOGÍAS ÁGILES DE DESARROLLO	10
2.4.1 METODOLOGÍA DE DESARROLLO PROGRAMACIÓN EXTREMA.....	11
2.4.2 CARACTERÍSTICAS DE LA PROGRAMACIÓN EXTREMA	13
2.4.3 FASES DE DESARROLLO DEL PROYECTO XP	15
2.4.4 FASE DE EXPLORACIÓN.....	16
2.4.5 FASE DE PLANIFICACIÓN	18
2.4.6 FASE DE ITERACIÓN.....	19
2.4.7 FASE DE PRODUCCIÓN	26
2.5 SOFTWARE LIBRE	27
2.6 LENGUAJE UNIFICADO DE MODELADO – UML.....	28
2.6.1 DIAGRAMAS DE CASOS DE USO	29
2.6.2 DIAGRAMAS DE CLASE.....	31
2.7 MÉTRICAS DE CALIDAD.....	32
2.7.1 ESTÁNDARES ISO/IEC 9126 CALIDAD DE PRODUCTOS SOFTWARE	33
CAPÍTULO III MARCO APLICATIVO.....	37

3.1	INTRODUCCIÓN.....	38
3.2	VALORES	38
3.2.1	COMUNICACIÓN	38
3.2.2	CORAJE	38
3.2.3	SIMPLICIDAD.....	38
3.2.4	CONTINUO SEGUIMIENTO.....	39
3.3	PRINCIPIOS.....	39
3.3.1	RETROALIMENTACIÓN A ESCALA FINA	39
3.3.2	PROCESO CONTINUO EN LUGAR DE POR LOTES.....	39
3.3.3	ENTENDIMIENTO COMPARTIDO	40
3.3.4	BIENESTAR DEL PROGRAMADOR	40
3.4	DESCRIPCION DE LOS ACTORES DEL NEGOCIO	40
3.5	DESCRIPCION DE LOS ESCENARIOS.....	42
3.6	FASE DE EXPLORACIÓN.....	44
3.6.1	HISTORIA DE USUARIO.....	44
3.6.2	DETALLE DE LAS HISTORIAS DE USUARIO	46
3.7	FASE DE PLANIFICACIÓN.....	48
3.7.1	ESTIMACIÓN DE ESFUERZO	49
3.7.2	PLANIFICACIÓN	50
3.8	ITERACIONES	52
3.8.1	PRIMERA ITERACIÓN.....	52
3.8.2	SEGUNDA ITERACIÓN	55
3.8.3	TERCERA ITERACIÓN	62
3.8.4	CUARTA ITERACIÓN	71
3.9	DIAGRAMA DE DESPLIEGUE	75
CAPÍTULO IV MÉTRICAS DE CALIDAD		76
4.1	INTRODUCCIÓN.....	77
4.2	FUNCIONALIDAD	77
4.3	CONFIABILIDAD	77
4.4	USABILIDAD	78
4.5	EFICIENCIA	78
4.6	FACILIDAD DE MANTENIMIENTO	79
4.7	PORTABILIDAD	79

4.8	INTEROPERABILIDAD	79
4.9	SEGURIDAD	79
CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES		81
5.1	CONCLUSIONES	82
5.2	RECOMENDACIONES.....	82
BIBLIOGRAFÍA.....		84
ANEXO A: MANUAL DE USUARIO		85
ANEXO B: HISTORIAS DE USUARIO.....		100

ÍNDICE DE TABLAS

Tabla 3. 1 Descripción de escenarios para el registro de oferta laboral	42
Tabla 3. 2 Descripción de escenario para la búsqueda de ofertas laborales	42
Tabla 3. 3 Descripción de escenario para búsqueda de postulantes.....	43
Tabla 3. 4 Descripción de escenario para el registro de postulación.....	43
Tabla 3. 5 Historia de Usuario: Registro de Institución.....	44
Tabla 3. 6 Tarea: Diseñar estructura de Datos.....	45
Tabla 3. 7 Tarea: Registro de Datos de la institución.....	45
Tabla 3. 8 Tarea: Registro del Encargado de la Institución	46
Tabla 3. 9 Instituciones	49
Tabla 3. 10 Demanda Laboral.....	49
Tabla 3. 11 Administración de Postulantes	49
Tabla 3. 12 Reportes	49
Tabla 3. 13 Administrador	50
Tabla 3. 14 Planificación de iteraciones.....	50

ÍNDICE DE FIGURAS

Figura 2. 1 Capas de la Ingeniería de software.....	9
Figura 2. 2 Métodos con mayor agilidad	12
Figura 2. 3 Enfoque tradicional frente a programación extrema	12
Figura 2. 4 Prácticas de la Programación Extrema	14
Figura 2. 5 Ciclo de vida de un proyecto desarrollado con Programación Extrema.....	16
Figura 2. 6 Formato de Historias de Usuario.....	17
Figura 2. 7 Factores para el desarrollo del proyecto	19
Figura 2. 8 Desarrollo Fase de Iteración	20
Figura 2. 9 Flujo de actividades en el Desarrollo.....	20
Figura 2. 10 Formato de tarjeta CRC	22
Figura 2. 11 Propiedad Colectiva de código.....	25
Figura 2. 12 Ejemplo de diagrama de caso de uso	30
Figura 2. 13 Ejemplo de diagrama de clases	32
Figura 3. 1 Diagrama de casos de uso general.....	41
Figura 3. 2 Cronograma de desarrollo de iteraciones.....	51
Figura 3. 3 Diagrama de Registro de Institución	52
Figura 3. 4 Pantalla de Registro de Institución	53
Figura 3. 5 Pantalla de registro de encargado de institución	53
Figura 3. 6 Registro de Oferta Laboral.....	55
Figura 3. 7 Pantalla de Registro de la Demanda Laboral	56
Figura 3. 8 Pantalla de Búsqueda de Ofertas Laborales	57
Figura 3. 9 Pantalla de Búsqueda de Ofertas Laborales	58
Figura 3. 10 Pantalla para mostrar detalle de la oferta.....	58
Figura 3. 11 Pantalla para editar Oferta Laboral	60
Figura 3. 12 Pantalla para la eliminación de una oferta laboral	60
Figura 3. 13 Registro de Postulante.....	62
Figura 3. 14 Pantalla para capturar datos complementarios	63
Figura 3. 15 Pantalla para capturar conocimientos o aptitudes del postulante	63
Figura 3. 16 Pantalla para registrar la experiencia laboral del postulante.....	64
Figura 3. 17 Pantalla para registrar los cursos realizados del postulante.....	64
Figura 3. 18 Ventana de búsqueda rápida	66

Figura 3. 19 Ventana de búsqueda avanzada.....	66
Figura 3. 20 Ventana detalle del Postulante.....	67
Figura 3. 21 Pantalla para actualizar datos complementarios	68
Figura 3. 22 Pantalla para capturar conocimientos o aptitudes del postulante	69
Figura 3. 23 Pantalla para registrar la experiencia laboral del postulante.....	69
Figura 3. 24 Pantalla para registrar los cursos realizados del postulante	70
Figura 3. 25 Ventana que muestra datos estadísticos.....	71
Figura 3. 26 Muestra los usuarios registrados.....	73
Figura 3. 27 Modificación de acceso al sistema	73
Figura 3. 28 Cambio de contraseña a los usuarios	73
Figura 3. 29 Diagrama de Despliegue del sistema	75

CAPÍTULO I

MARCO REFERENCIAL

1.1 INTRODUCCIÓN

El trabajo¹ es una necesidad que tiene cada individuo al momento de culminar los estudios realizados en una institución de educación superior, luego buscar un lugar donde poder aplicar todos los conocimientos adquiridos de manera que pueda resolver los problemas que se presenten. Dentro de esta búsqueda se presentan situaciones en las que el empleo escasea, ya sea por la gran cantidad de profesionales que salen al mercado laboral en busca del mismo objetivo o porque no encuentran formas de promoción.

El empleo puede obtenerse de forma directa o indirecta, dependiendo de dónde se busca un trabajo. Partiendo de la forma en la que uno se presenta a un trabajo, ya sea por una convocatoria, por un aviso en el periódico o un contacto entre un amigo. La Bolsa de Trabajo es variada, por los diferentes rubros que existen y los requisitos que se tienen al momento de contratar a un trabajador. En estos tiempos el avance de las tecnologías de información ayuda a conseguir ofertas de trabajo, como ser la Bolsa de Trabajo Virtual, que ayudan de manera significativa a encontrar oportunidades de trabajo.

Estas Bolsas de Trabajo Virtual donde ofertan y/o demandan, se pueden encontrar de varias formas como ser las públicas, orientadas a todo el público en general, las privadas que están orientadas a un sector en especial.

En el caso de los profesionales que salen de la Carrera de Informática, que al momento de terminar los estudios superiores, no se cuenta con una herramienta tecnológica, donde se pueda dar oportunidad de ofrecer sus servicios como profesionales, tanto en entidades públicas, como privadas. Es ahí donde se quiere resolver, el problema de encontrar trabajo, antes o después de terminar la carrera. Ya que el mercado necesita de profesionales capaces que puedan ejercer sus habilidades en distintos ámbitos.

Cada día se descubren nuevas bolsas de trabajo surgidas en la red, y es lógico, ya que Internet es un entorno de comunicación global y un espacio apropiado para que empresas y profesionales puedan resolver sus necesidades.

¹ Esfuerzo personal para la producción y comercialización de bienes y/o servicios con un fin económico, que origina un pago en dinero o cualquier otra forma de retribución (OIT, 1919).

1.2 PROBLEMÁTICA

Durante varios años, el problema de conseguir trabajo, al terminar el pre-grado en la Carrera de Informática, era incierto, ya que todos esperan ejercer la profesión al cabo de un largo tiempo de estudio. Esto debido a que no hay un encaminamiento por parte de la carrera para acomodar a sus recién egresados a fuentes laborales o facilitarles algún mecanismo para ayudarles a conseguir una fuente laboral.

Esto puede ser, debido a los siguientes problemas:

- La falta de convenios interinstitucionales donde se requiera personal informático
- La falta de una herramienta, que pueda promover a los nuevos profesionales.
- La falta de oportunidad para los nuevos profesionales
- La falta de información de las cualidades de los nuevos profesionales
- La falta de un catálogo digital para buscar de currículos de los nuevos profesionales.
- La falta de información de las tendencias del mercado laboral.
- La falta de capacitación de los nuevos profesionales en ramas de especialización
- La falta de información confiable sobre los cualidades de los nuevos profesionales
- La falta de información sobre las demandas específicas en el mercado laboral

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Desarrollar un Sistema de Información de bolsa de trabajo para los egresados de la carrera de Informática.

1.3.2 OBJETIVOS SECUNDARIOS

Se tienen los siguientes objetivos secundarios:

- Desarrollar un módulo para registrar a los egresados
- Desarrollar un módulo de búsquedas de egresados
- Desarrollar un módulo para registrar especialidades de los egresados
- Desarrollar un módulo para registrar instituciones
- Desarrollar un módulo para publicar ofertas laborales
- Desarrollar un módulo para publicar tendencias laborales

1.4 JUSTIFICACIÓN

1.4.1 JUSTIFICACIÓN SOCIAL

El sistema de información coadyuva a encontrar oportunidades de trabajo para los nuevos profesionales y disminuir la falta de empleo al momento de egresar de la Carrera de Informática.

1.4.2 JUSTIFICACIÓN ECONÓMICA

El sistema ayuda, a reducir costos al momento de buscar oportunidades de empleo y promocionar a los nuevos profesionales. En la estimación de costos se emplea el método COCOMO, como sigue:

Datos:

LDC = 1500 (Líneas de código)

Sueldo = 500 \$/m

El Esfuerzo

ED = 2.4 (KLDC) 1.05 h-m → $2.4 (1.5) 1.05 = 3.78 \text{ h-m} = 4 \text{ h-m}$

Tiempo de desarrollo

TD = 2.5 (ED) 0.38 m → $TD = 2.5 (15) 0.38 \text{ m} = 3.8\text{m}$

Productividad

PR = LDC / ED → $PR = 1500 / 4 = 375 \text{ LDC/h-m}$

Nº médio de personas

$$PE = ED / TD \quad h \rightarrow PE = 4 / 3.8 \quad h = 1.05 = 1 \text{ h}$$

Calculando el costo de Proyecto

$$\text{Costo por LDC} = 1500 / 375 = 4 \text{ \$/LDC}$$

$$\text{Costo total del proyecto} = 1500 * 4 = 6000 \text{ \$}$$

1.5 ESTADO DEL ARTE

1.5.1 TRABAJOS SIMILARES

Luego de una revisión bibliográfica en la Carrera de Informática se pudo evidenciar la existencia de algunos proyectos relacionados al actual proyecto, como ser:

“SISTEMA DE INFORMACIÓN CLIENTE/SERVIDOR PARA LOS PROCESOS DE LA BOLSA DE TRABAJO” (Calderón Rodríguez, 2004), este proyecto se realizó en el Ministerio de Trabajo, cuyo objetivo es almacenar información necesaria de la Oferta y Demanda laboral para agilizar el proceso de colocación. Este trabajo realiza las labores de registro de la oferta, registro de la demanda, colocación del trabajador en general, colocación de trabajadora del hogar, emisión de reportes, llegando a ser las principales tareas.

“SISTEMA AUTOMATIZADO DE LA BOLSA DE TRABAJO DECENTE” (Gutierrez Salas, 2004), este proyecto se realizó para el Gobierno Municipal de El Alto, donde el objetivo del proyecto es almacenar información de la cadena laboral gestionando el envío y seguimiento laboral, de esta manera proporcionar información de valor a los usuarios y entidades participantes del vínculo laboral, generando condiciones necesarias para la elaboración y toma de decisiones más acertadas.

Por otro lado, a nivel externo se encontró otros trabajos relacionados al trabajo actual:

“SISTEMA DE INFORMACIÓN PARA BOLSA DE TRABAJO” (Gómez Garduza, 2010), este proyecto se realizó en la universidad de Veracruz, el objetivo de este trabajo es Desarrollar un sistema de información que permita a los egresados de los tres programas de estudio que imparte la Facultad de Contaduría y Administración, vincularse con el mercado laboral.

“PROYECTO E-WORK – DISEÑO DE UN SISTEMA DE INFORMACIÓN FACILITADOR EN LA COLOCACIÓN DE ESTUDIANTES Y/O PROFESIONALES DEL INSTITUTO DE CIENCIAS HUMANÍSTICAS Y ECONÓMICAS EN EL SECTOR EMPRESARIAL, UTILIZANDO TECNOLOGÍA INTERNET” (Parodi Ledezma & Ruiz Jaramillo, 2005), El presente trabajo se realizó en la Escuela Superior Politécnica del Litoral Quito Ecuador donde se plantea una ayuda al CEPROEM – ICHE (Centro de Prácticas Profesionales y Empleo) en el proceso de colocación de estudiantes del Instituto de Ciencias Humanísticas y Económicas en plazas de empleo para la realización de las prácticas empresariales (requisito indispensable de graduación), mediante un sistema que utiliza tecnología Internet, así como mediante el planteamiento de un formato restringido de currículum, para obtener una discriminación positiva.

1.6 METODOLOGÍA

Para el desarrollo del Proyecto se utilizó una metodología Orientada a Objetos. Se usará Programación Extrema (XP). El modelado de análisis y diseño del sistema se realizó con UML².

1.7 TÉCNICAS Y HERRAMIENTAS

A continuación se detalla las técnicas y herramientas utilizadas, como ser:

- Framework CodeIgniter para PHP
- Framework JQuery para JavaScript.
- CSS, para la maquetación de las páginas
- PostgreSQL como gestor de Base de Datos
- Apache como servidor web
- Ubuntu Server para la implementación
- Enterprise Architect para el diseño de UML
- Netbeans, como IDE de programación
- Deziign For Database, para el diseño de la base de datos
- EMS SQL Manager for Postgres, para el manejo de la base de datos.

² Unified Modeling Language (Lenguaje Unificado de Lenguaje)

CAPÍTULO II

MARCO TEÓRICO

2.1 INTRODUCCIÓN

En el presente capítulo se describen y explican los conceptos que se aplican para el desarrollo del proyecto como base teórica, la metodología y las herramientas que se utilizaron.

2.2 BOLSA DE TRABAJO

La bolsa de trabajo son aquellas empresas especializadas en buscar y dotar personal a las entidades solicitantes, ya sea con personal calificado o no calificado. La característica principal de estas oficinas es que abastecen personal que cumple con los requerimientos del solicitantes, de acuerdo a especificaciones del puesto de trabajo, con la garantía de eficiencia y eficacia en el trabajo laboral.

Las bolsas de trabajo, son una más de las tantas maneras que existen en la actualidad, para buscar trabajo, ya que son justamente para aquello, para lo que han sido creadas.

Originalmente, las bolsas de trabajo, fueron diseñadas por las universidades, para buscar empleo para sus egresados. Cada facultad, poseía o posee, una base de datos, con los nombres de las personas egresadas de sus aulas. Por ende, las ponía en contacto, con aquellas empresas que buscaban personal, correspondiente a esa carrera. De igual manera, las universidades publican aquellos anuncios, que intentan captar profesionales, que hayan salido, de aquellas carreras requeridas por las empresas.

En la actualidad, con la arremetida de Internet, las bolsas de trabajo se han tomado la red. Ya que existen diversas páginas, que ofician de bolsas de trabajo. Son aquellas páginas, en las cuales las empresas ofrecen diversos cargos, según la materia de conocimiento de los postulantes. Un creciente número de personas, recurren a las mismas, para conseguir una fuente laboral. Las bolsas de trabajo más amplias, ya que cualquier persona puede acceder a sus ofertas.

Las bolsas de trabajo prestan un valioso servicio, para aquellas personas que están en la búsqueda de un puesto laboral. Asimismo, aquellas personas que desean cambiar de trabajo, también pueden hallar aquel que sea de su gusto.

Por lo mismo, es que se puede decir, que las bolsas de trabajo operan como un gran mercado laboral. Donde existen oferentes y demandantes. Todos los cuales, buscan una mejor perspectiva laboral a futuro (Sauceda Gastélum, 2008).

2.3 INGENIERIA DE SOFTWARE

La ingeniería de Software es: 1) La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operaciones y mantenimiento de software; 2) El estudio de enfoques (Pressman, 2010).

La ingeniería de software es una tecnología con varias capas. Como se aprecia en la siguiente Figura 2.1.



Figura 2. 1 Capas de la Ingeniería de software
Fuente: (Pressman, 2010)

Cualquier enfoque de ingeniería (incluso la de software) debe basarse en un compromiso organizacional con la calidad. La administración total de la calidad, Six Sigma y otras filosofías similares alimentan la cultura de mejora continua, y es esta cultura la que lleva en última instancia al desarrollo de enfoques cada vez más eficaces en la ingeniería de software. El fundamento en que se apoya la ingeniería de software es el compromiso con la calidad.

El fundamento para la ingeniería de software es la capa proceso. El proceso de ingeniería de software es el aglutinante que une las capas de la tecnología y permite el desarrollo racional y oportuno del software de conjunto. El proceso define una estructura que debe establecerse para la obtención eficaz de la tecnología de ingeniería de software. El proceso de software forma la base para el control de la administración de proyectos de software y establece el contexto en el que se aplican métodos técnicos, se generan productos del trabajo (modelos,

documentos, datos, reportes, formatos, etc.), se establecen puntos de referencia, se asegura la calidad y se administra el cambio de manera apropiada (Pressman, 2010).

2.4 METODOLOGÍAS ÁGILES DE DESARROLLO

El software en la mayoría de los casos es caótico y muchas veces caracterizada por frases como “codifica y corrige”. El código se escribe muchas veces con un plan en mente y el diseño se va armando con decisiones a corto plazo.

Como respuesta a la burocracia de las metodologías tradicionales, surgen las metodologías ágiles de desarrollo, con diferencia inmediata de que estas metodologías en contraste a las tradicionales no están orientadas al documento sino al código, sin embargo este aspecto no es relevante de las metodologías ágiles de desarrollo.

Las metodologías ágiles son adaptables en lugar de predicativas, a diferencia de las metodologías tradicionales que planifican la mayor situación de casos, es así que las metodologías tradicionales se resisten a aceptar cambios posteriores al diseño; para las metodologías ágiles el cambio es bienvenido, ya que se intentan crear procesos que crezcan en el tiempo y espacio.

Las metodologías ágiles son orientadas a los recursos humanos y no a los procesos, una premisa de la que parte las metodologías tradicionales se basa en que un proceso funcionará bien con cualquier persona que los use, en cambio las metodologías ágiles sostienen que ningún proceso podrá disfrazar las habilidades del grupo de desarrollo (Beck, 2000).

Dentro de las metodologías ágiles de desarrollo de software más representativas podemos citar:

- XP (Extreme Programming)
- La familia de Crystal de Cickburn
- ASD (Desarrollo de Software Adaptable de highsmith)
- Scrum
- DSDM (Método de desarrollo de Sistemas Dinámico)

2.4.1 METODOLOGÍA DE DESARROLLO PROGRAMACIÓN EXTREMA

La programación Extrema o XP, es una metodología ágil de desarrollo de software, centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

En la Tabla 2.1 se puede apreciar la comparación entre los diferentes métodos ágiles que existen, y sus respectivas valoraciones con respecto a los criterios de comparación.

Tabla 2. 1 Los valores más altos representan una mayor agilidad

Fuente: Extraído de (Wells, 2009)

INDICE DE AGILIDAD					
CRITERIOS DE COMPARACIÓN	ASD	Crystal	DSDM	Scrum	XP
Sistema como algo cambiante	5	4	3	5	5
Colaboración	5	5	4	5	5
Características Metodológicas(CM)					
Resultados	5	5	4	5	5
Simplicidad	4	4	3	5	5
Adaptabilidad	5	5	3	4	3
Excelencia técnica	3	3	4	3	4
Prácticas de colaboración	5	5	4	4	5
Media CM	4,4	4,4	3,6	4,2	4,4
Media Total	4,8	4,5	3,6	4,7	4,8

De acuerdo a los resultados representados en la Tabla 2.1, se concluye que los tres primeros métodos con mayor agilidad y de acuerdo a la escala, son XP, SCRUM y ASD (ver Figura 2.2):

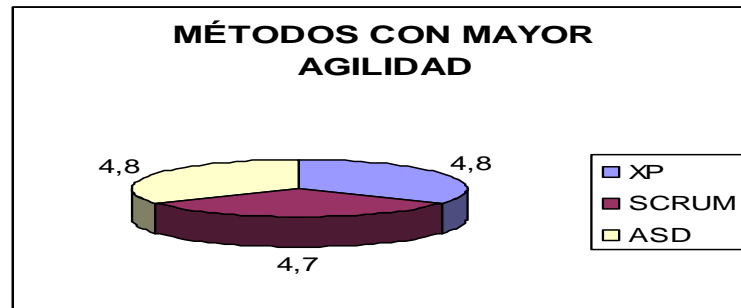


Figura 2. 2 Métodos con mayor agilidad

Fuente: (Beck, 2000)

De acuerdo a Kent Beck todo en el software cambia, los requisitos cambian, el diseño cambia, el negocio cambia, la tecnología cambia, el equipo cambia, los miembros del equipo cambian, el problema es la incapacidad de adaptarse al cambio cuando éste tiene lugar. (Beck, 2000)

La metodología de desarrollo Programación Extrema es una técnica que permite al equipo de desarrollo organizar, planificar y elaborar un software de manera incremental tomando en cuenta siempre la calidad ante todo aplicando pruebas y permitiendo cambios de manera constante sin elevar el costo excesivamente en términos de tiempo y dinero al desarrollar el proyecto, (ver Figura 2.3).



Figura 2. 3 Enfoque tradicional frente a programación extrema

Fuente: (Beck, 2000)

2.4.2 CARACTERÍSTICAS DE LA PROGRAMACIÓN EXTREMA

Las características aplicadas en esta metodología son la base fundamental para su desarrollo. A continuación se describen (Fowler & Beck, 2000):

- **Comunicación:** Los programadores están en constante comunicación con los clientes para satisfacer sus requisitos y responder rápidamente a los cambios de los mismos. Existen muchos problemas que surgen en los proyectos se deben a que después de concretar los requisitos que debe cumplir el programa no hay una revisión de los mismos, pudiendo dejar olvidados puntos importantes.
- **Simplicidad:** Codificación y diseños simples. Muchos diseños son tan complicados que cuando se quieren ampliar resulta bastante difícil hacerlo o se tienen que desechar y partir de cero.
- **Retroalimentación:** Mediante la realimentación se ofrece al cliente la posibilidad de conseguir un sistema apto a sus necesidades ya que se va mostrando el proyecto a tiempo para ser cambiado y retroceder a una fase anterior para rediseñarlo al gusto del cliente.
- **Valentía:** Se debe tener valentía o coraje para cumplir los tres puntos anteriores; Hay que tener valor para comunicarse con el cliente y enfatizar algunos puntos, a pesar de que esto pueda dar la sensación de ignorancia por parte del programador, hay que tener valentía para mantener un diseño simple y no optar por el camino más fácil y por último hay que tener valor y confiar en que la realimentación sea efectiva.

Se tiene un conjunto de doce principios básicos los cuales forman parte de las cuatro características mencionadas que son necesarias para la correcta aplicación de la Programación Extrema. Es decir, son buenas prácticas de desarrollo de software que se fueron perfeccionando a lo largo del tiempo, y que en esta metodología son aplicadas con frecuencia (Beck, 2000).

A continuación se describen los pasos de las prácticas de Programación Extrema (ver Tabla 2.2).

Tabla 2. 2 Prácticas de la Programación Extrema

Fuente: (Beck, 2000)

Valores	Nro. De Principio	Principios y practicas XP
Comunicación	1	Integración Continua
	2	Refactorización
	3	Entregas pequeñas
Simplicidad	4	Diseño simple
	5	Metáfora
	6	Propiedad colectiva del código
	7	Estándar de codificación
Retroalimentación	8	El principio de las pruebas
	9	Proceso de planificación
	10	El cliente en el sitio
	11	Programación en parejas
Valentía	12	La semana de 40 horas

La Programación Extrema está compuesta por una serie de prácticas y actividades. Se agrupan en tres distintos niveles; el nivel más básico muestra las prácticas diarias y constantes, el segundo nivel muestra las prácticas más generales, y el tercer nivel muestra prácticas para todo el equipo aplicables a lo largo del proyecto de manera general.

La Figura 2.4 refleja los puntos mencionados anteriormente.

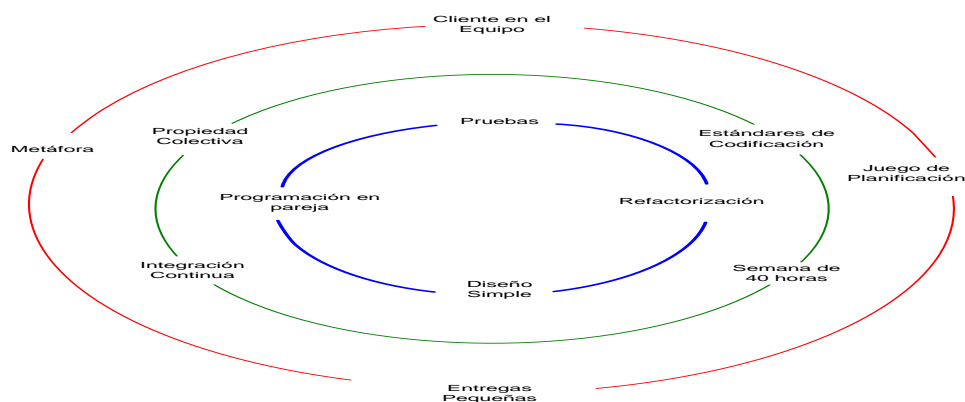


Figura 2. 4 Prácticas de la Programación Extrema

Fuente: (Beck, 2000)

Utilizando los valores y prácticas de XP, se da a conocer como programadores y clientes interactúan en el desarrollo del software haciendo uso de estas prácticas.

Es muy importante hacer uso de las prácticas y valores de Programación Extrema para obtener buenos resultados en el desarrollo de software. La Tabla 2.3 muestra a los participantes de cada práctica donde son diferenciadas en prácticas conjuntas, prácticas de programador, prácticas gerenciales y prácticas del cliente.

Tabla 2. 3 Interacción de participantes de cada práctica

Fuente: (Hurtado, 2004)

Prácticas Conjuntas	Interacción (una fase del desarrollo)
Práctica de Programador	Desarrollo orientado a pruebas Programación en pares Refactorización Propiedad Colectiva Interacción Continua Diseño Simple Estándares de Codificación
Prácticas Gerenciales	Todo el equipo en el mismo lugar "Cliente en el sitio" Ritmo sostenible
Prácticas de Cliente	Narración de historias (una tarea continua del cliente) Planeamiento de entrega Entregas frecuentes

2.4.3 FASES DE DESARROLLO DEL PROYECTO XP

La metodología Programación Extrema se divide en 4 fases principales: exploración, planificación, iteraciones y la producción. Estas 4 fases principales conforman el ciclo de vida de un proyecto desarrollado con Programación Extrema.

La Figura 2.5 detalla el ciclo de vida de un proyecto desarrollado con XP, detallando las fases que tiene.

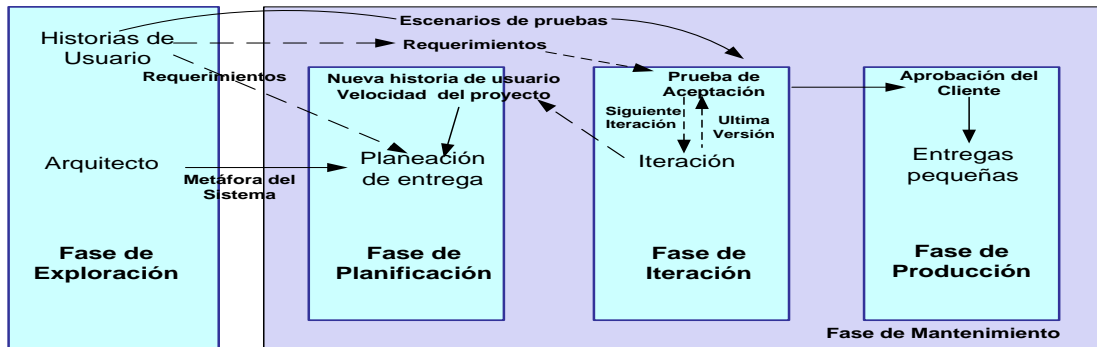


Figura 2. 5 Ciclo de vida de un proyecto desarrollado con Programación Extrema.

Fuente: (Beck, 2000)

En la primera fase de exploración se realiza la definición de los requerimientos iniciales con los que se comenzará el proyecto, después en la fase de planificación se priorizan las tareas más importantes a ser ejecutadas, es decir se planifica el desarrollo del proyecto, y se estiman los tiempos iniciales de desarrollo, luego en la fase de iteración se desarrolla el software en sí, esta fase junto con la de planificación y de producción siguen un ciclo incremental repetitivo, que se repite n veces de acuerdo a la magnitud del sistema, se alcanza la conclusión del proyecto cuando el cliente ya no tiene ningún requerimiento y está conforme con el software desarrollado

2.4.4 FASE DE EXPLORACIÓN

En esta fase se realizan el análisis de requerimientos inicial con el que el proyecto se inicia para su desarrollo, se elaboran las historias de usuarios (HU) teniendo a todo el equipo reunido, para entender las historias de usuario; el usuario elabora dichas historias con ayuda del equipo de desarrollo para que estén enfocadas al desarrollo del sistema de forma concreta y concisa. Es muy importante obtener historias de usuario que expresen las necesidades específicas de forma que sea comprensible por el grupo de desarrollo.

Durante la exploración se define la arquitectura del sistema, escogiendo las herramientas tecnológicas que posibilitaran la construcción del sistema de manera óptima y eficiente.

Se debe explorar las limitaciones tecnológicas en cuanto a hardware y software, también se deben estimar de manera superficial pero muy conscientemente el tiempo de desarrollo de cada tarea principal. El encargado de diseño debe establecer una metáfora para el sistema, que será manejado a lo largo del proyecto por el cliente y por el equipo de desarrollo.

El desarrollo de esta fase debe cumplir con los siguientes elementos (ver Figura 2.6)

Historias de usuario (HU)

Historia de Usuario	N: 1	8
Nombre:	2	Prioridad: 3
Historia:	4	
Objetivo:	5	
Estimación:	6	Dependencia: 7

1 Número de historias de usuario
 2 Nombre de la historia de usuario
 3 Prioridad en planificación de entregas
 4 Descripción de la historia de usuario
 5 Objetivo de la historia de Usuario
 6 Estimación del tiempo de desarrollo
 7 Dependencia del Programador
 8 Nombre del sistema

Figura 2. 6 Formato de Historias de Usuario

Fuente: (Beck, 2000)

Es el primer paso de cualquier proyecto que sigue la metodología XP, consiste en definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso en el paradigma orientado a objetos, pero con algunas diferencias. Constan de 3 ó 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles; no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados, etc.

Son usadas para estimar tiempos de desarrollo del segmento de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que

especifica la historia de usuario. Cuando llega la hora de **implementar una historia de usuario, el cliente y los desarrolladores se reúnen para** concretar y detallar lo que tiene que hacer dicha historia. El tiempo ideal para el desarrollo de una historia de usuario es entre 1 y 3 semanas (Beck, 2000).

Metáfora

Una metáfora es una historia compartida que describe como debe funcionar el sistema. El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo (Beck, 2000).

2.4.5 FASE DE PLANIFICACIÓN

El propósito de esta fase es encontrar un acuerdo entre los clientes y el equipo de desarrollo en cuanto a fechas y que historias de usuarios serán desarrolladas primero, para el desarrollo de esta etapa se debe definir el plan de entregas en el cual se establecen prioridades y tiempos de desarrollo de acuerdo al alcance, recursos, tiempo y calidad, estas son cuatro variables para el desarrollo de cualquier proyecto, el alcance, recurso, tiempo son tres variables para el desarrollo que se modifican generalmente en la planificación, no siendo así con la cuarta que es la calidad, entonces toda la planificación se basa en equilibrar las tres variables para la planificación correcta y así no afectar la calidad, entonces toda la planificación se basa en equilibrar las tres variables para la planificación correcta y así no afectar la calidad.

Planeación de Entregas: tras definir las historias de usuarios es necesario crear un plan de entregas, o publicaciones donde se indican las historias de usuario que se implementarán para cada versión de la aplicación, también las fechas en las que se publicarán dichas versiones, esto se efectúa estimando el tiempo de cada historia de usuarios.

Para crear un plan de entregas se debe realizar una estimación del tiempo de desarrollo de acuerdo a las historias de usuarios que se tienen, y a partir de esto indicar el tiempo estimado de la iteración. La prioridad para el desarrollo de las historias de usuarios, y el número de iteraciones estimado requerido para cada historia de usuario (Beck, 2000).

Después de la elaboración de un plan de entregas se deben tener claros los siguientes cuatro factores para el desarrollo del proyecto (ver Figura 2.7).

● Objetivos: los objetivos que se deben cumplir (son principalmente las historias que se deben desarrollar en cada versión)
● Tiempo: el tiempo que tardarán en desarrollarse y publicarse las versiones de la aplicación
● Involucrados: define el número de personas que trabajarán en el desarrollo.
● Evaluación de la calidad: define cómo se evaluará la calidad del trabajo realizado.

Figura 2. 7 Factores para el desarrollo del proyecto

Fuente: (Beck, 2000)

La duración de esta fase tiene una duración de uno a tres días. En cuanto al diseño de base de datos se lo analizará en la fase de iteración, solo se diseñará e implementaran las tablas y columnas que utilizan las historias de usuario en la iteración correspondiente, esto no se especifica en el plan de entregas, ni en las historias de usuario.

2.4.6 FASE DE ITERACIÓN

Para una primera iteración se debe escoger las historias de mayor importancia, esto conduce a crear el esquema completo de todo el sistema. Es importante tener en cuenta el objetivo u objetivos principales de la iteración, si algún objetivo cambia por decisión del cliente, entonces deberá ser replanteado en la fase de Planificación de Entregas. (Fowler & Beck, 2000).

Al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en la “Planeación de entregas” que serán implementadas. También se seleccionan las historias de usuario que no pasaron el test de aceptación que se realizó al terminar la última iteración, en caso de ser la primera iteración no se toma en cuenta este aspecto.

Las entradas para la fase de iteración son el plan de entregas, las fallas y la proyección de una siguiente iteración, determinadas por las historias de usuario, las fallas en la prueba de aceptación de una iteración anterior, y la velocidad del proyecto respectivamente. A partir de

estas nuevas entradas se elabora un plan de iteración y se procede al desarrollo. La Figura 2.8 muestra esta fase de forma detallada respecto a las entradas, salidas, su comportamiento y funcionamiento (Wells, 2009).

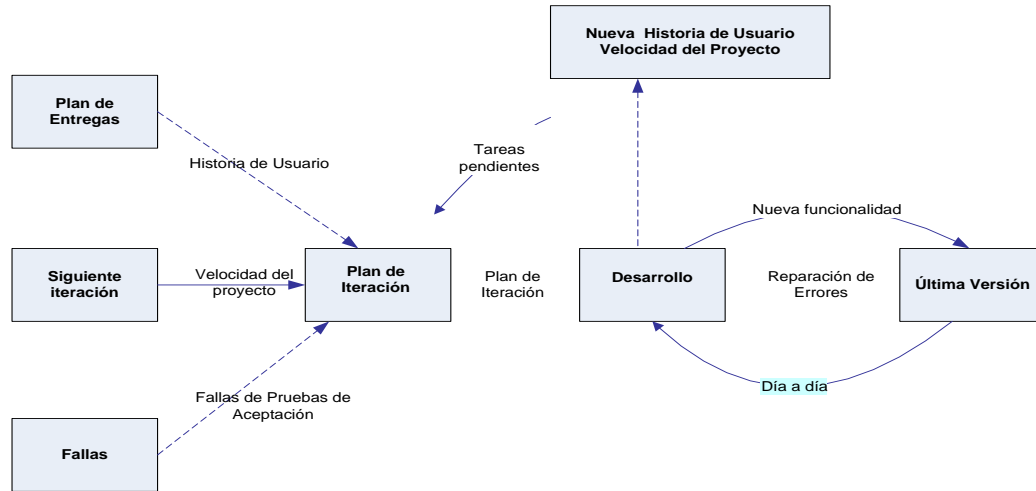


Figura 2. 8 Desarrollo Fase de Iteración

Fuente: (Wells, 2009)

Después de elaborar el plan de iteración se procede al desarrollo, que tiene como entradas el mismo plan de iteración, el desarrollo cotidiano definido por las tareas a realizarse y los test de aceptación fallidos, entonces se procede a reuniones diarias, que lleva al desarrollo en si (la codificación), a través de la prioridad de código colectiva detallada más adelante, todo este proceso se muestra en la Figura 2.9.

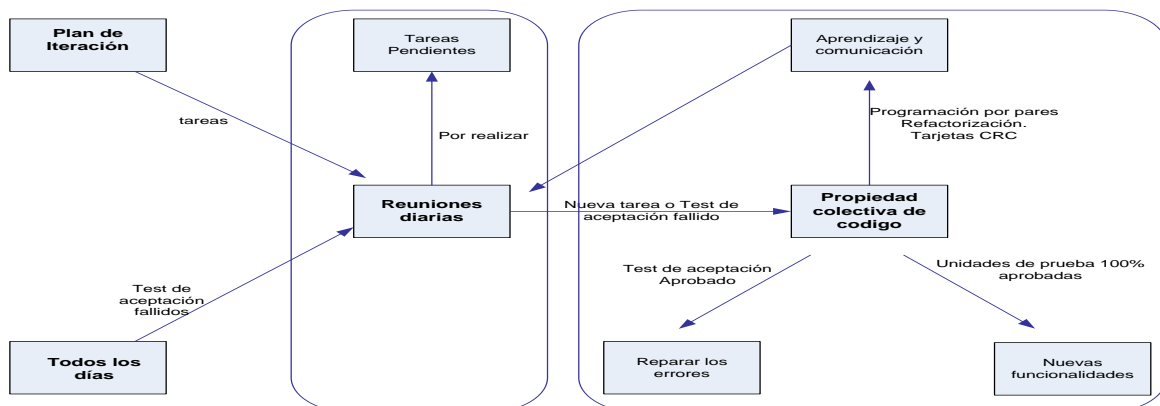


Figura 2. 9 Flujo de actividades en el Desarrollo

Fuente: (Wells, 2009)

Para lograr un buen desarrollo se deben definir historias de usuarios y dividir las en tareas de entre 1 y 3 días de duración cada una, que se asignaran a los programadores, el desarrollo de cada iteración en la asignación y estimación de las historias de usuario. Este proceso tiene mucha relación con el plan de iteración.

Las historias de usuarios seleccionadas se distribuyen en el Plan de Entregas asignando a cada par de programadores una historia, y si son muy grandes se las divide para el desarrollo de cada iteración se debe tomar en cuenta los siguientes elementos y prácticas: (ver Tabla 2.4).

Tabla 2. 4 Elementos y prácticas de utilidad

Fuente: (Wells, 2009)

ELEMENTOS Y PRÁCTICAS	Tarjetas CRC
	Diseño simple
	Eliminar funcionalidad extra
	Refactorizar
	Velocidad del proyecto
	Cliente siempre disponible
	Código con estándares
	Unidad de prueba
	Programación en pareja
	Integración frecuente
	Propiedad colectiva de código
	Reuniones diarias
	40 horas semanales

- **Tarjeta CRC.** La utilidad de las tarjetas CRC (Clase, Responsabilidad y Colaboración) ayudan al equipo a definir actividades durante el diseño del sistema. Cada tarjeta representa una clase en la programación orientada a objetos y define sus responsabilidades (lo que ha de hacer) y las colaboraciones con las otras clases (como se comunica con ellas) (Beck, 2000).

El formato de la tarjeta CRC se da de la siguiente forma: La clase a la que pertenece el objetivo se escribe en la parte superior de la tarjeta, en una columna a la izquierda se escriben las responsabilidades un objetivo que debe cumplir el objeto, a la derecha las clases que colaboran con cada responsabilidad.

El formato de las tarjetas se lo muestra en la Figura 2.10, y cuyas medidas son de 10 cm. X 15 cm. (Beck, 2000).



Figura 2. 10 Formato de tarjeta CRC

Fuente: (Fowler & Beck, 2000)

El objeto del tamaño de las tarjetas CRC es escribir solo lo indispensable de manera que entre cómodamente y se mantenga la simplicidad. Su diseño permite desprenderse del método de trabajo basado en procedimientos y trabajar con una metodología basada en objetivos, y que el equipo completo contribuya en la tarea de diseño (Beck, 2000).

- **Diseño simple.** La Programación Extrema (XP) exige conseguir diseños simples y sencillos, para conseguir un diseño fácil de entender e implementar que a la larga su desarrollo implicará menos tiempo y esfuerzo.

El plan correcto para el diseño simple de software en cualquier momento es:

- Pasar todas las pruebas
- No debe haber duplicados lógicos, ni duplicación oculta
- Mínimo número de clases y métodos.

Cada pieza del diseño en el sistema tiene que justificar su existencia. Un diseño simple permite menos tiempo de desarrollo que uno complejo; mantener la simplicidad en el diseño evitando agregar funcionalidades extra (Hurtado, 2004).

- **Eliminar funcionalidad extra.** Nunca se debe añadir funcionalidad extra al programa aunque se piense que en un futuro será utilizada. Solo el 10 % de la misma es utilizada. Se debe centrar la atención en la tarea que se ha fijado para la jornada de trabajo y ejecutarla lo mejor posible (Hurtado, 2004).
- **Refactorizar.** Significa mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Supone revisar de nuevo la codificación, para procurar optimizar su funcionamiento (Beck, 2000)

Las entradas al proceso de refactorizar son códigos complejos, que tiene como salida códigos simples.

Es común reutilizar código ya creado que suele contener funcionalidad que no son utilizadas, esto es un grave error porque existe un riesgo de generar código inestable y mal diseñado, por este motivo es necesario refactorizar para reutilizar un código. Implica mantener el código limpio fácil de comprender, modificar y ampliar. Esto resulta un poco costoso al principio, pero resulta fundamental para realizar diseños futuros.

- **Velocidad.** La velocidad del proyecto es una medida que representa la rapidez con la que se desarrolla el mismo, estimarla es muy sencillo, basta con contar el número de historias de usuario o CRC que se implementan en una iteración, de esta forma se sabrá el cupo de historias que se desarrollan en las distintas iteraciones.

Con la velocidad del proyecto se controlará si todas las tareas se desarrollan en el tiempo del que dispone la iteración. Es conveniente reevaluar esta medida cada 3 o 4 iteraciones y si se aprecia que no es adecuada hay que negociar con el cliente un plan de Entregas.

- **Cliente siempre disponible.** Un pilar importante para la programación extrema (XP) es tener al usuario siempre disponible. No solo para ayudar al equipo de desarrollo, comunicación con el usuario, preferiblemente cara a cara, en persona y sin intermediarios. Durante la reunión del plan de entregas, el usuario propondrá que las historias de usuario se incluyan en cada plan.

También se negociaran los plazos de entrega. El usuario o cliente tomará las decisiones que le afecten para alcanzar los objetivos de su negocio (Beck, 2000)

También es necesario que el cliente colabore en la realización de los test. Estos test comprueban que el sistema está listo para pasar a la fase de producción. El usuario comprobará los resultados obtenidos y tomará decisiones en cuanto a la utilización o del sistema realizado.

- **Se debe escribir código de acuerdo a los estándares.** El código debe ser desarrollado siguiendo los estándares de desarrollo para facilitar su lectura y modificación por cualquier miembro del equipo de desarrollo. Es decisivo, para plantear con éxito la propiedad colectiva del código. Se usa la metáfora y el glosario de términos descritos en la fase de exploración como base para estándares de código.
- **Desarrollar primero la unidad de pruebas.** Cuando los test son creados antes que el código, la implementación del código se agiliza. El tiempo empleado en desarrollar un test y algo de código para probarlo es aproximadamente el mismo tiempo que se emplea en crear exclusivamente dicho código. La creación de las unidades de test ayudan al programador a tener una visión acerca de cómo se implementará dicha tarea, y del comportamiento del programa. Esta manera de trabajar resulta especialmente beneficiosa en el diseño de complicados sistemas software (Beck, 2000).
- **Programación en pareja.** Todo el código que formará parte del plan, será desarrollado por dos personas que trabajan de forma conjunta en un ordenador, de esta manera se incrementa la calidad del software desarrollado sin afectar al tiempo

de entrega. Se parte sabiendo que el equipo de dos personas posee unos conocimientos similares en cuanto a la tarea que se realiza, están aproximadamente al mismo nivel. Mientras uno de ellos se encarga de pensar las estrategias que permiten llevar dichas tácticas a su máximo exponente, ambos roles son intercambiables.

- **Integración frecuente.** Los programadores deben actualizar rápidamente los módulos desarrollados con versiones recientes del trabajo realizados, de esta manera todo el equipo trabaja siempre con la última versión. Esta integración se llevará a cabo cuando el éxito en las pruebas para su test correspondiente sea total, o cuando se trate de una parte que constituye un todo funcional, en cuanto esta se termine.

- **El código es común a todos.** Esta filosofía permite que cualquier miembro del equipo contribuya al desarrollo de cualquier parte del proyecto. Cualquier programador tiene la autorización de cambiar una línea de código para añadir funcionalidad o eliminar algún fallo, siempre que beneficie a la arquitectura y pase todas las pruebas. A partir de las tarjetas CRC se crean las unidades de prueba, luego se procede a la programación que es por parejas, mientras se refactorizará, elementos y prácticas. Existe un proceso continuo de integración, que conlleva a la superación de las unidades de prueba y luego a la aceptación, (ver Figura 2.11).

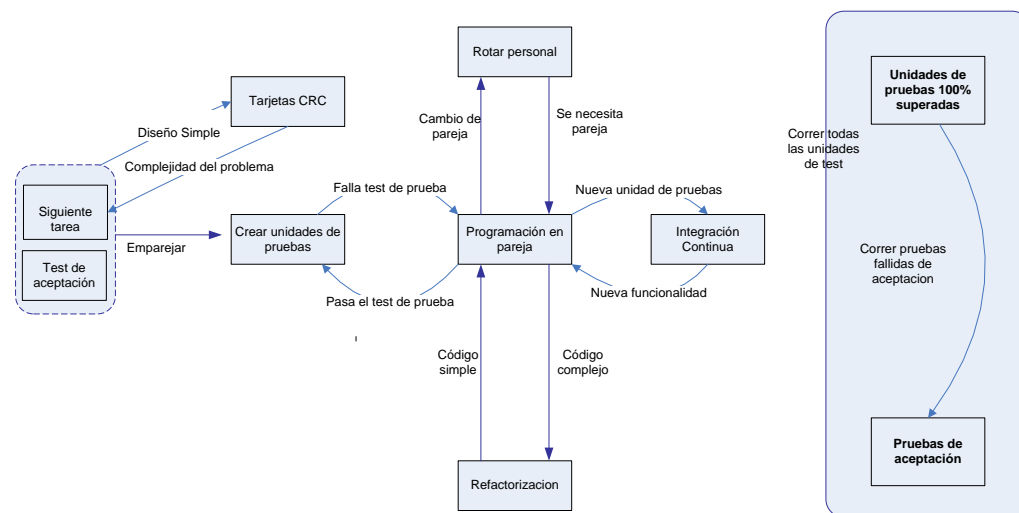


Figura 2. 11 Propiedad Colectiva de código

Fuente: (Fowler & Beck, 2000)

- **40 Horas semanales.** Trabajar horas extras absorbe el espíritu y la motivación del equipo. Aquellos proyectos que requieren horas extras para la culminación se convierten en un problema (Fowler & Beck, 2000).

Es una mala idea incorporar nueva gente al proyecto, una vez que este ya ha comenzado. Se trabajará un máximo de 40 horas semanales. Nadie es capaz de trabajar 60 horas a la semana y hacerlo con calidad. (Beck, 2000).

- **Reuniones diarias.** Es necesario que los desarrolladores se reúnan diariamente y expongan los problemas que presentan soluciones e ideas de forma conjunta. Las reuniones tienen que ser fluidas y todos tienen voz y audio.

2.4.7 FASE DE PRODUCCIÓN

Su duración debe ser de un día o menos, se debe llegar periódicamente a esta etapa para comprobar el respectivo avance del proyecto, en esta fase se realiza la producción del sistema, se toman los siguientes elementos como fundamentales para la conclusión de la misma (Beck, 2000).

- **Código de unidad de prueba.** Las unidades de test o pruebas constituyen un pilar básico de la programación Extrema, un error cometido frecuentemente es pensar que se construyen los test en los últimos meses de la realización de un proyecto. Las unidades de test están directamente relacionadas con el concepto de posesión de código. En cierta manera una parte del código no será reemplazado si no supera los test que existen para ese código.

Después de cada modificación, se debe emplear los test para verificar que un cambio en la estructura no introduce un cambio en la funcionalidad. Si se añade nuevas capacidades al código, se tiene que rediseñar la unidad de test, para adaptarse a la nueva funcionalidad. De esta manera, la probabilidad de que exista un fallo en ambos (test y código) es menor.

- Aprobación de unidades de pruebas antes de implementación. Las unidades de test son incluidas junto con el código dentro del repositorio. El código no se considera completo si este no consta de su unidad de test correspondiente. El código es implementado cuando supera las correspondientes unidades de test.
- Ejecución de pruebas de aceptación frecuentes. Las pruebas de aceptación son creadas a partir de las historias de usuario. Durante una iteración la historia de usuario seleccionada en la planificación de iteraciones, se convierte en una prueba de aceptación. El cliente o usuario especifica los aspectos a testear cuando una historia de usuario ha sido correctamente implementadas. Una historia de usuario tiene más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento. Una prueba de aceptación es como una caja negra. Cada una de ellas representa una salida esperada del sistema. Es responsabilidad del cliente verificar la corrección de las pruebas de aceptación y tomar decisiones acerca de las mismas.

Una historia de usuario no se considera completa hasta que no supera las pruebas de aceptación. Esto significa que debe desarrollarse un nuevo test de aceptación para cada iteración o se considerará que el equipo de desarrollo no realiza ningún progreso. La puntualidad de las pruebas de aceptación debe ser pública para todo el equipo. La garantía de calidad es una parte esencial. La realización de este tipo de pruebas y la publicación de los resultados debe ser lo más rápido posible, para que los desarrolladores realicen con la mayor rapidez posible los cambios que sean necesarios. A las pruebas de aceptación también se las conoce con el nombre de pruebas de funcionalidad, y constituyen la garantía de que los requerimientos fijados por los usuarios han sido reflejados en el sistema (Beck, 2000).

2.5 SOFTWARE LIBRE

El software libre se refiere a la libertad de los usuarios para ejecutar, copiar distribuir, estudiar, cambiar y mejorar el software, esta clasificación con lleva a cuatro libertades (Stallman, 2004).

- Libertad 1: libertad para ejecutar el programa sea cual sea el propósito.
- Libertad 2: libertad para estudiar el funcionamiento del programa y adaptarlo a necesidades del usuario.
- Libertad 3: libertad de redistribuir copias y ayudar así al vecino.
- Libertad 4: libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad.

Software libre es cualquier programa cuyos usuarios gocen de estas libertades. De modo que se debe ser libre de redistribuir copias con o sin modificaciones de forma gratuita o cobrando por distribución. Gozar de esta libertad significa, entre otras cosas no tener que pedir permiso ni pagar para ello. Asimismo se debe ser libre para introducir modificaciones y utilizarlas de forma privada, ya sea en su trabajo o en su tiempo libre, sin siquiera mencionar su existencia. Si se decide publicar estos cambios, no deberías estar obligado a notificárselo a ninguna persona ni de ninguna forma en particular (Stallman, 2004).

El software libre no significa que sea “no comercial”. Cualquier programa libre estará disponible para su uso, desarrollo y distribución comercial. El desarrollo comercial del software libre ha dejado de ser excepcional y de hecho ese software libre comercial es muy importante. Las normas sobre el empaquetamiento de una versión modificada son perfectamente aceptables siempre que no restrinjan efectivamente la libertad para publicar versiones modificadas. Por la misma razón, serán igualmente aceptables aquellas normas que establezcan que “sí distribuyo el programa de esta forma, deberás distribuirlo de la misma manera”, cabe destacar que esta norma permite decidir si publicar o no el programa.

También hay que admitir la posibilidad de que una licencia exija enviar una copia modificada y distribuida de un programa a su desarrollador original (Stallman, 2004).

2.6 LENGUAJE UNIFICADO DE MODELADO – UML

El lenguaje unificado de modelado es una técnica de modelado de objetos y como tal supone para especificar, visualizar, construir y documentar artefactos de un sistema de software. Es un sistema rotacional (que entre otras cosas, incluye el significado de sus notaciones) destinado a los sistemas (Larman, 2004).

UML es una consolidación de muchas de las notaciones y permite a los creadores de sistemas realizar diseños y modelos que interpreten el mundo real de manera fácil y comprensible. UML no es una notación cerrada es genérica, extensible y configurable de acuerdo a los diseños realizados, es un lenguaje gráfico que permite modelar, construir y documentar los elementos que se forman un sistema de software (Muller, 2000).

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se conoce como modelo. Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

A continuación se describen brevemente los diagramas de UML que se emplearon en la documentación.

2.6.1 DIAGRAMAS DE CASOS DE USO

Un caso de uso representa una unidad funcional coherente de un sistema, subsistema o clase, en un caso de uso uno o más actores interactúan con el sistema que realiza algunas acciones.

La mejor forma de desarrollar un buen diagrama de caso de uso es mediante entrevista directa con los usuarios o posibles futuros usuarios del sistema, poniendo atención a cada una de las actividades o pasos que se van a ir desarrollando desde un primer momento hasta un momento final.

La elaboración de diagrama de casos de uso ayuda poderosamente a una analista a comprender la forma en que un sistema deberá comportarse, obteniendo los requerimientos desde el punto de vista del usuario.

En todo caso de uso siempre hay un actor, que es quien inicia, luego otro actor (que puede ser el mismo que inicia el caso de uso o puede ser otro diferente), que recibirá algo por parte del sistema. La representación gráfica es directa, tal como se muestra en la Figura 2.12, donde se detalla cómo se representa a los actores dentro de un contexto.

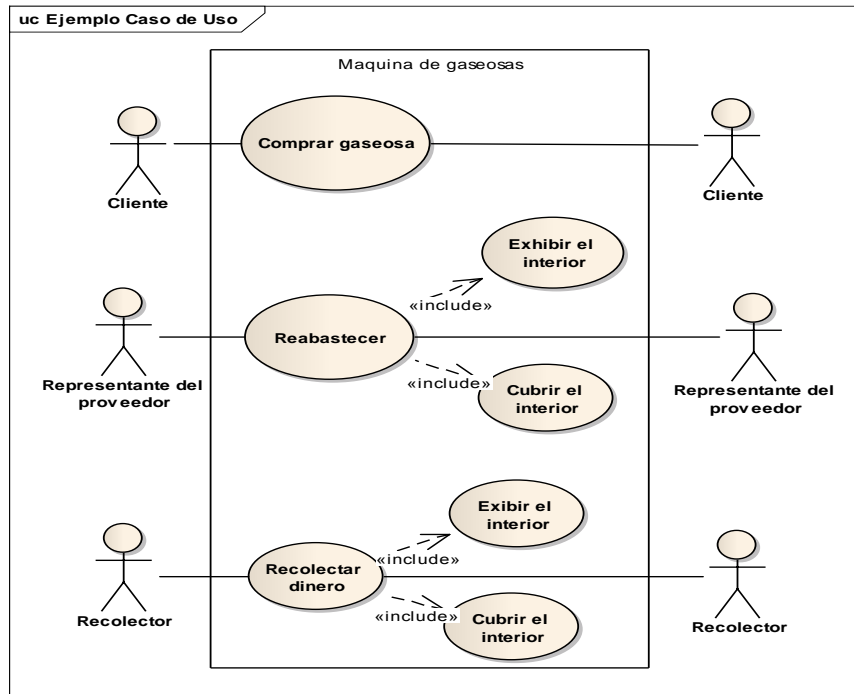


Figura 2. 12 Ejemplo de diagrama de caso de uso

Fuente: (Quinceno, 2011)

En la Figura 2.12, la elipse representa el caso de uso. Las dos figuras en los extremos izquierdo y derecho son los actores que intervienen. El actor que inicia se encuentra a la izquierda del caso de uso y el que recibe a la derecha. El nombre del actor aparece justo debajo de él, y el nombre del caso de uso aparece, ya sea dentro de la elipse o justo debajo de ella. Una línea asociativa conecta a un actor con el caso de uso, y representa la comunicación entre el actor y el caso de uso. La línea asociativa es sólida. El rectángulo envuelve a los caso de uso dentro del sistema.

Un aspecto importante es la inclusión, que permite volver a utilizar los pasos de un caso de uso dentro de otro. Se utiliza líneas discontinuas con una punta de flecha que se dirige hacia las clases dependientes. Este tipo de flechas con el símbolo que representa inclusión en la nomenclatura UML. Justo sobre la línea se agrega una palabra que se utiliza para revelar la relación de inclusión existente entre los casos de uso vinculados. Esta palabra debe ponerse precisamente sobre la flecha (o a un lado de la línea, si esta fuera vertical o incluso debajo de la línea si el espacio de trabajo obliga a ello) y debe ser bordeada por dos pares de paréntesis angulares. La palabra que se ha utilizado es incluir.

La extensión, consiste en añadir un nuevo caso de uso al caso original, que sería el caso de uso base (Schmuller, 2002).

2.6.2 DIAGRAMAS DE CLASE

Los diagramas de clases colaboran en lo referente al análisis. Permiten al analista hablarles a los clientes en propia terminología, lo cual hace posible que los clientes indiquen importantes detalles de los problemas que requieren ser resultados.

Antes que nada, un objeto es instancia de una clase (o categoría). Usted y yo, por ejemplo, somos instancias de la clase Persona. Un objeto cuenta con la estructura, es decir atributos (propiedades) y acciones. Las acciones son todas las actividades que el objeto es capaz de realizar. Los atributos y acciones en conjunto, se conocen como características o rasgos.

Como objetos de la clase Persona, usted y yo contamos con los siguientes atributos: altura, peso y edad (puede imaginar muchas más). También realizamos las siguientes tareas: comer, dormir, leer, escribir, hablar, trabajar, etcétera. Si tuviéramos que crear un sistema que manejara información acerca de las personas (como una nómina o un sistema para el departamento de recursos humanos). Sería muy probable que incorporáramos algunos de sus atributos y acciones en nuestro software.

Una clase tiene otro propósito, además de la categorización. En realidad es una plantilla para fabricar objetos.

- **Abstracción.** La abstracción se refiere a quitar las propiedades y acciones de un objeto para dejar solo aquellas que sean necesarias.
- **Especialización.** El proceso en el cual se definen las subclases.
- **Herencia de atributos:** las subclases heredan todos los atributos de la superclase.
- **Relaciones específicas:** aquellas que se aplican solo en las subclases.

En UML, un rectángulo es un símbolo que representa a la clase, y se divide en tres áreas. El área superior contiene el nombre de la clase, el área central contiene los atributos o propiedades, y el área inferior, las acciones, procedimientos, métodos o funciones. Un

diagrama de clases está formado por varios rectángulos de ese tipo conectados por líneas que muestran la manera en que las clases se relacionan entre sí.

En el ejemplo que se presenta en la Figura 2.13, se muestran tres clases, donde las líneas continuas con extremos en forma de triángulo sin rellenar, tal como se había explicado anteriormente. Son los símbolos de asociación de generalización, que sirven para mostrar herencia de una clase a otra, donde el extremo del triángulo sin rellenar apunta hacia la clase base, y el extremo de la línea continua que no tiene dicho triángulo se encuentra la clase que hereda. En síntesis, la clase base **Vehículo** es de donde heredan sus propiedades y acciones las clases **Auto** y **Camioneta**, la cual imaginarlo en un escenario de la vida real, es completamente atinado y razonable.

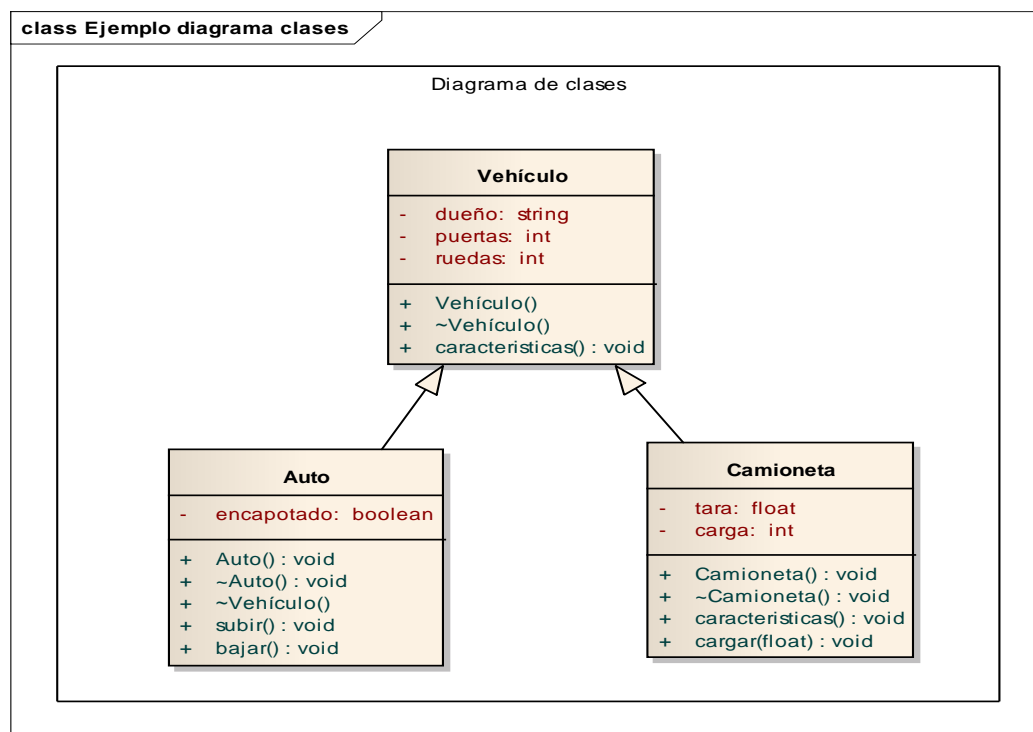


Figura 2. 13 Ejemplo de diagrama de clases

Fuente: (Schmuller, 2002)

2.7 MÉTRICAS DE CALIDAD

El objetivo no es necesariamente alcanzar una calidad perfecta, sino la necesaria y suficiente para cada contexto de uso a la hora de entrega y del uso por parte de los usuarios.

Es necesario comprender las necesidades reales de los usuarios con tanto detalle como sea posible (requisitos).

2.7.1 ESTÁNDARES ISO/IEC 9126 CALIDAD DE PRODUCTOS SOFTWARE

La ISO 9126 es un estándar internacional para la evolución del software. Esta supervisado por el proyecto SQuaRE, ISO 25000:2005, el cual sigue los mismos conceptos.

El estándar está dividido en cuatro partes la cuales dirigen, respectivamente lo siguiente: modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso.

El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características y sub características de la siguiente manera:

➤ **Funcionabilidad**

Adecuación. Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuarios específicos.

Exactitud. Capacidad del producto software para proporcionar los resultados o efecto concretos o acordados, con el grado necesario de precisión.

Interoperabilidad. Capacidad del producto software para interactuar con uno o más sistemas específicos.

Seguridad de Acceso. Capacidad del producto software para proteger información y datos de manera que las persona o sistemas no autorizados no puedan leerlos o modificarlos, al tiempo que no se deniega el acceso a las personas o sistemas autorizados.

Cumplimiento funcional. Capacidad del producto software para adherirse a normas, convenciones o regulaciones en leyes y prescripciones similares relacionadas con funcionabilidad.

➤ **Fiabilidad**

Madurez. Capacidad del producto software para evitar fallar como resultado de fallos en el software.

Tolerancia a fallos. Capacidad del producto software para mantener un nivel especificado de prestaciones en caso de fallos de software o de infringir sus interfaces especificados.

Capacidad de recuperación. Capacidad del producto software para restablecer un nivel de prestaciones especificado y de recuperar los datos directamente afectado en caso de fallo.

Cumplimiento de la fiabilidad. Capacidad del producto software para adherirse a normas, convenciones o regulaciones relacionadas con la fiabilidad.

➤ **Usabilidad**

Capacidad para ser entendido. Capacidad del producto software que permite al usuario entender si el software es adecuado y como puede ser usado para unas tareas o condiciones de uso particulares.

Capacidad para ser aprendido. Capacidad del producto software que permite al usuario aprender sobre su aplicación.

Capacidad para ser operado. Capacidad del producto software que permite al usuario operarlo y controlarlo.

Capacidad de atracción. Capacidad del producto software para ser atractivo al usuario.

➤ **Eficiencia**

Comportamiento temporal. Capacidad del producto software para proporcionar tiempo de respuesta, tiempo de proceso y potencia apropiados, bajo condiciones determinadas.

Utilización de recursos. Capacidad del producto software para usar las cantidades y tipos de recursos adecuado cuando el software lleva a cabo su función bajo condiciones determinadas.

Cumplimiento de la eficiencia. Capacidad del producto software para adherirse a normas o convenciones relacionadas con la eficiencia.

➤ **Mantenibilidad**

Capacidad para ser analizado. Es la capacidad del producto software para ser diagnosticadas deficiencias o causas de los fallos en el software, o para identificar las partes que van a ser modificadas.

Capacidad para ser cambiado. Capacidad del producto software que permite que una determinada modificación sea implementada.

Estabilidad. Capacidad del producto software para evitar efectos inesperados, debidos a modificaciones de software.

Capacidad para ser probado. Capacidad del producto software que permite que el software modificado sea validado.

Cumplimiento de la mantenibilidad. Capacidad del producto software para adherirse a normas o convenciones relacionadas con la mantenibilidad.

➤ **Portabilidad**

Adaptabilidad. Capacidad del producto software para ser adaptado a diferentes entornos, sin aplicar acciones o mecanismos distintos de aquellos proporcionados para este propósito por el propio software considerado.

Capacidad de Instalación. Capacidad del producto software para ser instalado en un entorno especificado.

Coexistencia. Capacidad del producto software para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes.

Capacidad para reemplazar. Capacidad del producto software para ser usado en lugar de otro producto software, para el mismo propósito, en el mismo entorno.

Cumplimiento de portabilidad. Capacidad del producto software para adherirse a normas o convenciones relacionadas con la portabilidad.

Cada subcaracterística (como adaptabilidad) está dividida en atributos. Un atributo es una entidad la cual puede ser verificada o medida en el producto software. Los atributos no están definidos en el estándar, ya que varían entre diferentes productos software.

Un producto software está definido en un sentido amplio como: los ejecutables, código fuente, descripciones de arquitectura, y así, como resultado, la noción de usuario se amplía tanto a operadores como a programadores, los cuales con usuarios de componentes como bibliotecas software.

El estándar provee un entorno para que las organizaciones definan un modelo de calidad para el producto software. Haciendo que se lleve a cada organización la tarea de especificar precisamente su propio modelo. Esto podría ser hecho, por ejemplo, especificando los objetivos para las métricas de calidad las cuales evalúan el grado de presencia de los atributos de calidad.

La calidad en las métricas de uso están solo disponibles cuando el producto final es usado en condiciones reales. Idealmente, la calidad interna determina la calidad externa y esta a su vez la calidad en el uso.

CAPÍTULO III

MARCO APLICATIVO

3.1 INTRODUCCIÓN

En el presente capítulo, en principio se presenta los lineamientos que tiene la metodología XP, en este apartado se describe las fases de: exploración, planificación, iteraciones con sus respectivas pruebas, hasta llegar a la fase de finalización del proyecto, y las correspondientes métricas de calidad apoyado con las herramientas UML para la documentación.

3.2 VALORES

Los cuatro valores que se deben tomar en cuenta según la metodología XP, para el desarrollo del software son:

3.2.1 COMUNICACIÓN

Se nutre con este valor mediante la comunicación directa con el encargado del proyecto, constantemente y cuando se tiene dudas, así como también los usuarios de otras áreas involucradas.

3.2.2 CORAJE

Durante el desarrollo del siguiente proyecto se expuso las dudas, para poder captar la idea de los requerimientos del usuario mediante las reuniones o comunicación directa con el personal del proyecto, debido a ello se tuvo que rehacer los puntos que no eran satisfactorios.

3.2.3 SIMPLICIDAD

La simplicidad del diseño se refleja en las tarjetas CRC del presente proyecto. Puede corroborarse en la documentación que el sistema tiene solo la funcionabilidad requerida en cada historia de usuario, no se incorporan funcionabilidades que no son requeridas en las historia de usuario.

3.2.4 CONTINUO SEGUIMIENTO

A través de las pruebas de usuario y aceptación, reuniones con el personal del proyecto, se mantiene una constante evaluación el desempeño del sistema, adicionando al mismo tiempo nuevas funcionalidades que se veía necesario en alguna de las iteraciones.

3.3 PRINCIPIOS

A continuación se presentan los doce principios de la metodología XP que en el caso de la programación extrema son llevados a tal punto que hace que el proyecto tenga éxito, y los cuales no son nuevos ya que son usados por otras metodologías.

3.3.1 RETROALIMENTACIÓN A ESCALA FINA

La retroalimentación a escala se describe en los siguientes puntos

- **El principio de pruebas:** Para cada historia de usuario se empleará pruebas de aceptación, las cuales se realizan de manera manual y cuyos resultados se reflejan en las respectivas iteraciones propuestas.
- **Proceso de planificación:** Este principio se describe en el punto 3.5 Fase II del presente capítulo, donde se detalla la planificación.
- **El cliente en el sitio:** Se implementa este principio, a través del desarrollo del proyecto en instalaciones de la Carrera de Informática, la cual hará el papel de cliente para la respectiva documentación.
- **Programación en parejas:** El presente trabajo lo realizara solamente una persona, a la vez será responsable de la documentación del mismo, pero mediante el constante intercambio de opiniones y trabajo conjunto con el personal encargado.

3.3.2 PROCESO CONTINUO EN LUGAR DE POR LOTES

- **Integración continua:** El sistema se ira integrando a medida que culmina el desarrollo de cada historia de usuario, pero antes de pasar la pruebas respectivas

- **Refactorización:** Las historias que no logran superar las pruebas respectivas se volverán a programar.
- **Entregas pequeñas:** Se realizan pequeñas entregas a la conclusión de cada historia de usuario para que se implanten y se entreguen. Los oficiales se empaquetan al final de cada iteración.

3.3.3 ENTENDIMIENTO COMPARTIDO

- **Diseño simple:** Al contar con la colaboración de los encargados del proyecto se mantiene la documentación del proyecto transparente y de fácil acceso a él, lo que hace que el diseño sea evidentemente simple.
- **Metáfora:** Se observan en la presentación de las tarjetas CRC que las metáforas utilizadas para el proyecto son sencillas.
- **Propiedad colectiva del código:** La propiedad del código es propiedad del autor.
- **Estándar de codificación:** Debido a que el proyecto es individual no se tuvo problemas con el estándar de codificación.

3.3.4 BIENESTAR DEL PROGRAMADOR

Semana de 40 horas: Promediando las horas de trabajo se cumple las 40 horas mencionadas.

3.4 DESCRIPCION DE LOS ACTORES DEL NEGOCIO

Los actores identificados que intervendrán en el negocio, se listan y describen a continuación:

- **Administrador:** Es la persona que administra el sistema, asigna permisos de accesos a usuarios según el rol que les corresponda, así como la atribución de eliminar datos.
- **Instituciones:** Usuarios que pueden ser ofertantes y demandantes laborales de acuerdo a sus requerimientos.

- **Postulante:** Esta persona ofertara sus aptitudes y conocimientos para ser reclutado por instituciones en busca de personal informático para trabajar.

La Figura 3.1 muestra los casos de uso general a utilizarse en el desarrollo de la aplicación donde destacan los actores que intervienen en el sistema y sus correspondientes procesos.

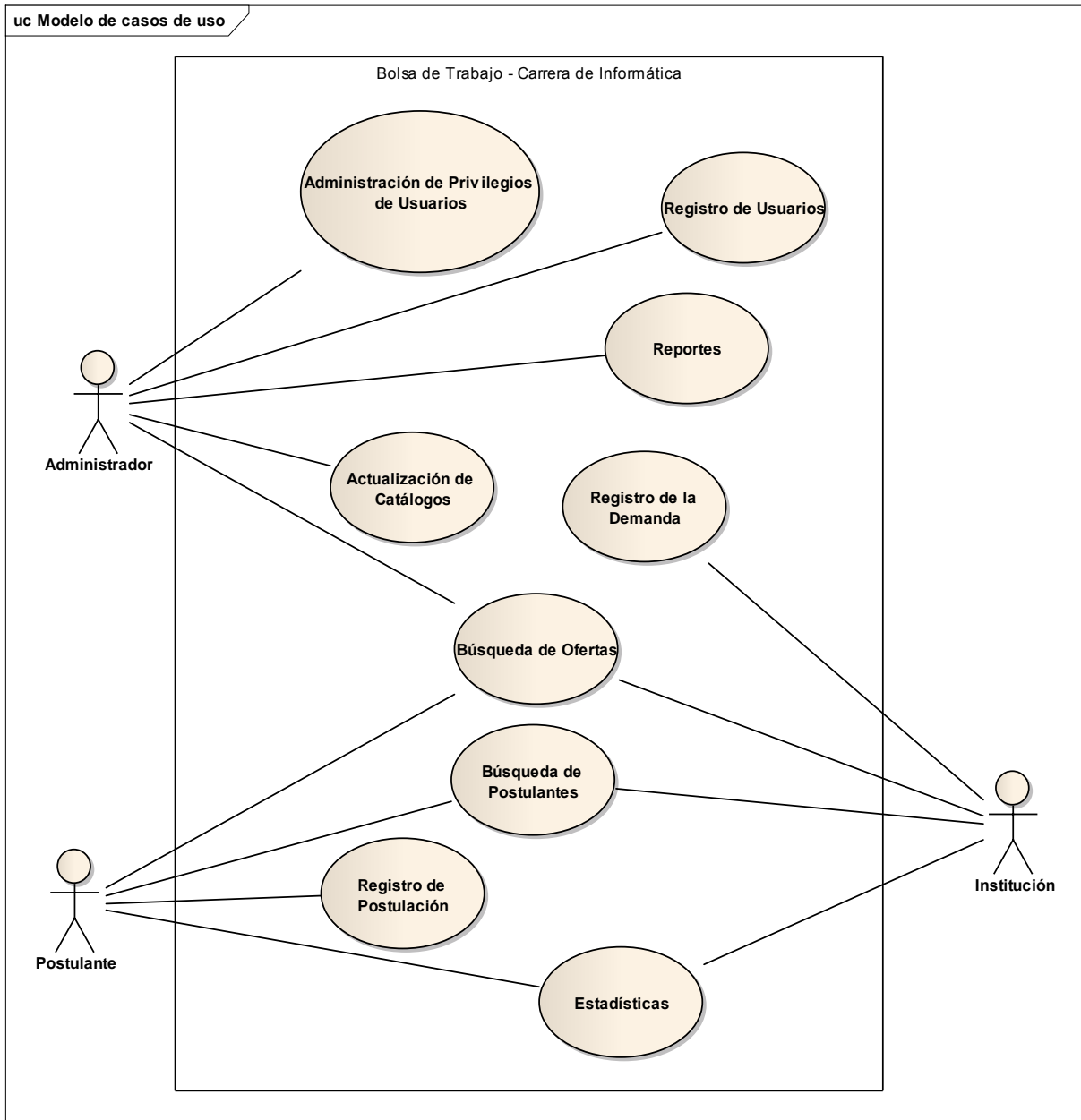


Figura 3. 1 Diagrama de casos de uso general

3.5 DESCRIPCION DE LOS ESCENARIOS

En este apartado se describen todos los escenarios que se presentaran en el desarrollo de la aplicación (ver Tabla 3.1, Tabla 3.2, Tabla 3.3 y Tabla 3.4).

Tabla 3. 1 Descripción de escenarios para el registro de oferta laboral

Título: Registro de la Demanda Laboral	Propósito: Registra los datos de la Demanda Laboral.
Utiliza a:	
Extiende a:	
Descripción: Captura los datos de la Demanda Laboral a publicarse	
Precondiciones: Ser usuario registrado	
Poscondiciones:	
Actores: Institución	
Episodios: <ol style="list-style-type: none"> 1. El usuario ingresa su login y password 2. El sistema verifica que los datos introducidos sean correctos 3. El sistema almacena en la base de datos 4. Finaliza el registro. 	

Tabla 3. 2 Descripción de escenario para la búsqueda de ofertas laborales

Título: Búsqueda de Ofertas Laborales	Propósito: Buscar ofertas laborales registradas en el sistema.
Utiliza a:	
Extiende a:	
Descripción: Busca oferta laborales en el sistema	
Precondiciones: Tener demanda laboral registrada	
Poscondiciones:	
Actores: Administrador, Institución. Postulante	
Episodios: <ol style="list-style-type: none"> 1. El usuario escribe parámetros de búsqueda de ofertas 2. El sistema verifica los datos introducidos para la búsqueda 3. El sistema busca en la base de datos las ofertas disponibles y las muestra 4. Fin de la búsqueda. 	

Tabla 3. 3 Descripción de escenario para búsqueda de postulantes

Título: Búsqueda de Postulantes	Propósito: Buscar Postulantes registrados en el sistema.
Utiliza a:	
Extiende a:	
Descripción: Busca Postulantes registrados en el sistema	
Precondiciones: Tener Postulantes registrados	
Poscondiciones:	
Actores: Administrador, Institución. Postulante	
Episodios: <ol style="list-style-type: none"> 1. El usuario escribe parámetros de búsqueda de postulantes 2. El sistema verifica los datos introducidos para la búsqueda 3. El sistema busca en la base de datos los postulantes registrados 4. El sistema despliega la lista de postulantes encontrados 5. Fin de la búsqueda. 	

Tabla 3. 4 Descripción de escenario para el registro de postulación

Título: Registro de Postulantes	Propósito: Registrar a los postulantes a la bolsa de trabajo.
Utiliza a:	
Extiende a:	
Descripción: Registra a los postulantes en el sistema	
Precondiciones: ser usuario del sistema	
Poscondiciones:	
Actores: Postulante	
Episodios: <ol style="list-style-type: none"> 1. El usuario ingresa su login y password 2. El usuario llena el formulario para postulantes 3. El sistema valida los datos obtenidos por el postulante. 4. El sistema captura y registra datos obtenidos, y los guarda en la base de datos. 5. El sistema despliega la información registrada. 6. Finaliza el registro. 	

3.6 FASE DE EXPLORACIÓN

En esta etapa de la exploración, que es la primera fase, se levantan las historias de usuarios y que se detallan a continuación así como también al mismo tiempo el desarrollador que hace de grupo de desarrollo se familiariza con la tecnología, herramientas y prácticas utilizadas en el transcurso del desarrollo del proyecto.

3.6.1 HISTORIA DE USUARIO

Las historias levantadas para el presente proyecto son:

➤ **Módulo de Instituciones**

El módulo se puede dividir en las siguientes historias de usuario y sus respectivas tareas (ver Tabla 3.5).

Tabla 3. 5 Historia de Usuario: Registro de Institución

Historia de Usuario	
Numero: 1	Usuario: Institución
Nombre Historia: Registro de Instituciones	
Prioridad del negocio: Alta	Riesgo en desarrollo: Alta (Alta/Media/Baja)
Puntos estimados: 2	Iteración asignada: 1
Descripción: Se registra lo datos de las instituciones, el encargado de la institución, quien estará a cargo de las publicaciones de la institución.	
Observaciones: Confirmado por el cliente.	

Tareas (ver Tabla 3.6, Tabla 3.7 y Tabla 3.8)

Tabla 3. 6 Tarea: Diseñar estructura de Datos

Tarea	
Numero: 1.1	Número historia: 1
Nombre Tarea: Diseñar una estructura de datos para registrar datos de la institución	
Tipo de tareas: Diseño	Puntos estimados: 1/2
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se realiza el diseño de la base de datos para el registro de datos de las instituciones.	

Tabla 3. 7 Tarea: Registro de Datos de la institución

Tarea	
Numero: 1.2	Número historia: 1
Nombre Tarea: Registro de datos de la institución	
Tipo de tareas: Desarrollo	Puntos estimados: 1
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrolló una ventana en la cual se introducen datos de la institución.	

Tabla 3. 8 Tarea: Registro del Encargado de la Institución

Tarea	
Numero: 1.3	Número historia: 1
Nombre Tarea: Registro de datos del encargado	
Tipo de tareas: Desarrollo	Puntos estimados: 1
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrolló una ventana en la cual se introducen datos del encargado de la institución.	

Las demás historias de usuario se encuentran en el ANEXO B.

3.6.2 DETALLE DE LAS HISTORIAS DE USUARIO

A continuación se realiza un listado general de las historias de usuario y sus respectivas tareas, las cuales se dividen en módulos, que se detallan en el punto anterior. De esta manera se pretende presentar un marco entendible de las historias de usuario, para así poder realizar posteriormente la planificación que presenta la metodología XP.

3.6.2.1 Módulo de Institución

En este módulo se describe la historia de usuario y sus tareas respectivas sobre el módulo de institución.

➤ **Historia de usuario 1:** Registro de Instituciones

Tarea 1: Diseñar una estructura de datos para registrar los datos de la institución

Tarea 2: Registro de datos de la institución

Tarea 3: Registro de datos del encargado

3.6.2.2 Módulo de la Demanda

En este módulo se describen las historias de usuario y sus tareas respectivas sobre el módulo de la Demanda.

- **Historia de usuario 2:** Registro de la Demanda
 - Tarea 1:** Diseñar una estructura de datos para registrar las ofertas laborales.
 - Tarea 2:** Registro de datos de las ofertas laborales

- **Historia de usuario 3:** Búsqueda de ofertas laborales
 - Tarea 1:** Búsqueda simple de Ofertas Laborales
 - Tarea 2:** Búsqueda avanzada de ofertas laborales
 - Tarea 3:** Mostrar detalle de oferta laboral

- **Historia de usuario 4:** Modificación de Ofertas Laborales
 - Tarea 1:** Edición de ofertas laborales
 - Tarea 2:** Eliminación de ofertas laborales

3.6.2.3 Módulo Postulantes

En este módulo se describen las historias de usuario y sus tareas respectivas sobre el módulo de Postulantes

- **Historia de usuario 5:** Registro de postulantes
 - Tarea 1:** Diseño de una estructura de datos para guardar los datos de los postulantes
 - Tarea 2:** Registro datos complementarios del postulante
 - Tarea 3:** Registro de conocimientos o aptitudes del postulante
 - Tarea 4:** Registro de experiencia laboral
 - Tarea 5:** Registro cursos realizados

- **Historia de usuario 6:** Búsqueda de Postulantes
 - Tarea 1:** Búsqueda simple de Postulantes
 - Tarea 2:** Búsqueda avanzada de Postulantes

Tarea 3: Mostrar detalle del Postulante

➤ **Historia de usuario 7:** Edición de datos Postulante

Tarea 1: Edición datos complementarios del postulante

Tarea 2: Edición de conocimientos o aptitudes del postulante

Tarea 3: Edición de experiencia laboral

Tarea 4: Edición cursos realizados

3.6.2.4 Módulo Reportes

En este módulo se describe la historia de usuario y sus tareas respectivas sobre el módulo de Reportes.

➤ **Historia de usuario 8:** Reportes estadísticos

Tarea 1: Reporte estadístico de postulantes con conocimientos

3.6.2.5 Módulo Administrador

En este módulo se describe la historia de usuario y sus tareas respectivas sobre el módulo de Administrador.

➤ **Historia de usuario 9:** Privilegios de usuarios

Tarea 1: Administrar privilegios de usuarios

Tarea 2: Cambio contraseña de usuarios

3.7 FASE DE PLANIFICACIÓN

La fecha de comienzo para el desarrollo del proyecto es el 17 de agosto de 2012. Se ha planeado cuatro iteraciones en las cuales se entregan diferentes versiones de los programas acordados. La cantidad y el orden de las iteraciones de historias de usuario así como la estimación de esfuerzo se detallan a continuación, de las cuales algunas de ellas han ido modificándose en el transcurso del desarrollo de proyecto y otras fueron eliminadas o reagrupadas debido a requerimientos u otros factores.

3.7.1 ESTIMACIÓN DE ESFUERZO

Se realiza la estimación de esfuerzo para los diferentes escenarios descritos anteriormente.

La Tabla 3.9 describe el esfuerzo de Instituciones

Tabla 3. 9 Instituciones

Historias de usuario	Pts.
Registro de Instituciones	1

La Tabla 3.10 describe el esfuerzo en Ofertas Laborales

Tabla 3. 10 Demanda Laboral

Historias de usuario	Pts.
Registro de Demanda Laboral	1
Búsqueda de Ofertas Laborales	1
Modificación de Ofertas Laborales	1

La Tabla 3.11 describe el esfuerzo de Administración de Postulantes

Tabla 3. 11 Administración de Postulantes

Historias de usuario	Pts.
Registro de Postulantes	1
Búsqueda de Postulantes	1
Edición de datos Postulante	1

La Tabla 3.12 describe el esfuerzo de los Reportes

Tabla 3. 12 Reportes

Historias de usuario	Pts.
Reporte Estadístico	1

La Tabla 3.11 describe el esfuerzo para la administración

Tabla 3. 13 Administrador

Historias de usuario	Pts.
Privilegios de usuario	1

3.7.2 PLANIFICACIÓN

Se realiza la planificación para las iteraciones que se realizaran, como se muestra a continuación (ver Tabla 3.14).

Tabla 3. 14 Planificación de iteraciones

ITERACIONES	Nro.	HISTORIAS	INICIO	FIN	OBSERVACIÓN
PRIMERA	1	Registro de Instituciones	13/08/2012	31/09/2012	
SEGUNDA	2	Registro de la Demanda Laboral	03/09/2012	07/09/2012	
	3	Búsqueda de Ofertas Laborales	10/09/2012	14/09/2012	
	4	Modificación de Ofertas Laborales	17/09/2012	21/09/2012	
TERCERA	5	Registro de Postulantes	24/09/2012	05/10/2012	
	6	Búsqueda de Postulantes	08/10/2012	12/10/2012	
	7	Modificación de datos Postulante	15/10/2012	26/10/2012	
CUARTA	8	Reportes	29/10/2012	02/11/2012	
	9	Privilegios de usuario	05/11/2012	16/11/2012	

La Figura 3.2 muestra el cronograma para las actividades que se planificaron a lo largo del proyecto.

Id.	Nombre	Comienzo	Fin	Duración	ago 2012				sep 2012				oct 2012				nov 2012						
					5/8	12/8	19/8	26/8	2/9	9/9	16/9	23/9	30/9	7/10	14/10	21/10	28/10	4/11	11/11	18/11	25/11		
1	Exploración	01/08/2012	10/08/2012	1,6s																			
2	Iteración 1	13/08/2012	31/08/2012	3s																			
3	Iteración 2	03/09/2012	21/09/2012	3s																			
4	Iteración 3	24/09/2012	26/10/2012	5s																			
5	Iteración 4	29/10/2012	16/11/2012	3s																			

Figura 3. 2 Cronograma de desarrollo de iteraciones

3.8 Iteraciones

En este apartado se desglosa las iteraciones que se planificaron en la Figura 3.2, de acuerdo a las fechas establecidas.

3.8.1 PRIMERA ITERACIÓN

La primera iteración, se realiza con la historia de usuario que se hizo para las instituciones.

Historia de usuario 1: Registro de instituciones

Tarea 1: Diseñar una estructura de datos para registrar los datos de la institución (ver Figura 3.3).

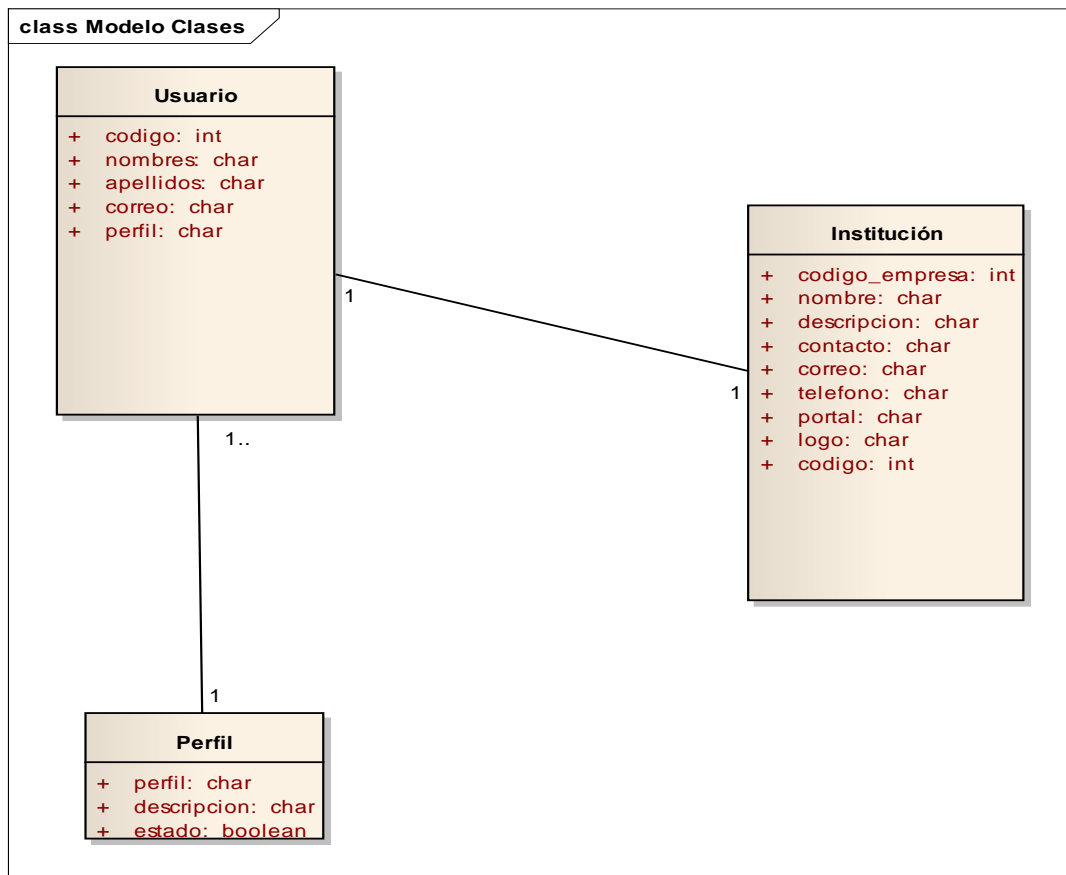


Figura 3. 3 Diagrama de Registro de Institución

Tarea 2: Registro de datos de la institución.

En esta pantalla se registra los datos de la Institución (ver Figura 3.4).



The screenshot shows a web form titled "Datos de la Empresa". It contains five text input fields stacked vertically. The first field is labeled "Nombre Empresa" and is empty. The second field is labeled "Descripción" and is empty. The third field is labeled "usuario@empresa.com" and contains a placeholder email address. The fourth field is labeled "Teléfono" and is empty. The fifth field is labeled "http://www.institución.com" and contains a placeholder URL. Below the input fields is a button labeled "Examinar...".

Figura 3. 4 Pantalla de Registro de Institución

Tarea 3: Registro de datos del encargado

En esta pantalla se registra los datos del encargado de la Institución (ver Figura 3.5).



The screenshot shows a web form titled "Datos del Encargado". It contains five text input fields stacked vertically. The first field is labeled "Nombre(s)" and is empty. The second field is labeled "Apellido(s)" and is empty. The third field is labeled "usuario@ejemplo.com" and contains a placeholder email address. The fourth field is labeled "Contraseña" and is empty. The fifth field is labeled "Confirmar Contraseña" and is empty. Below the input fields is a button labeled "Registrarse".

Figura 3. 5 Pantalla de registro de encargado de institución

Pruebas de aceptación para la historia 1

a. Identificar todos los posibles resultados observables de la historia

- Interfaz para el llenado de los datos de las instituciones.
- Pantalla para capturar los datos de la institución.
- Interfaz para el llenado de datos del encargado de la empresa
- Pantalla para capturar los datos del encargado de la institución.

b. Identificar los resultados que termina la historia y los que permiten continuar dentro de la historia.

- La historia termina cuando se termina de llenar los datos de la institución y del encargado de la misma.

c. Identificar los caminos de ejecución posibles

- Comienza la historia cuando se registra a la institución, al encargado con todos los datos.
- Valida los datos.
- La historia termina cuando completa los datos solicitados
- Almacena los datos en la base de datos.

d. Asigna un conjunto de valores válidos y valores del entorno a cada camino de ejecución para obtener el resultado esperado.

- Un conjunto de valores válidos está dado por el conjunto de datos compuesto por la información de la institución y del encargado.

e. Eliminación de caminos redundantes

- No existen caminos redundantes

3.8.2 SEGUNDA ITERACIÓN

La segunda iteración detalla el registro de la demanda laboral que la institución realizara una vez registrado.

Historia de usuario 2: Registro de Ofertas

Tarea 1: Diseñar una estructura de datos para registrar las demanda laboral (ver Figura 3.6)

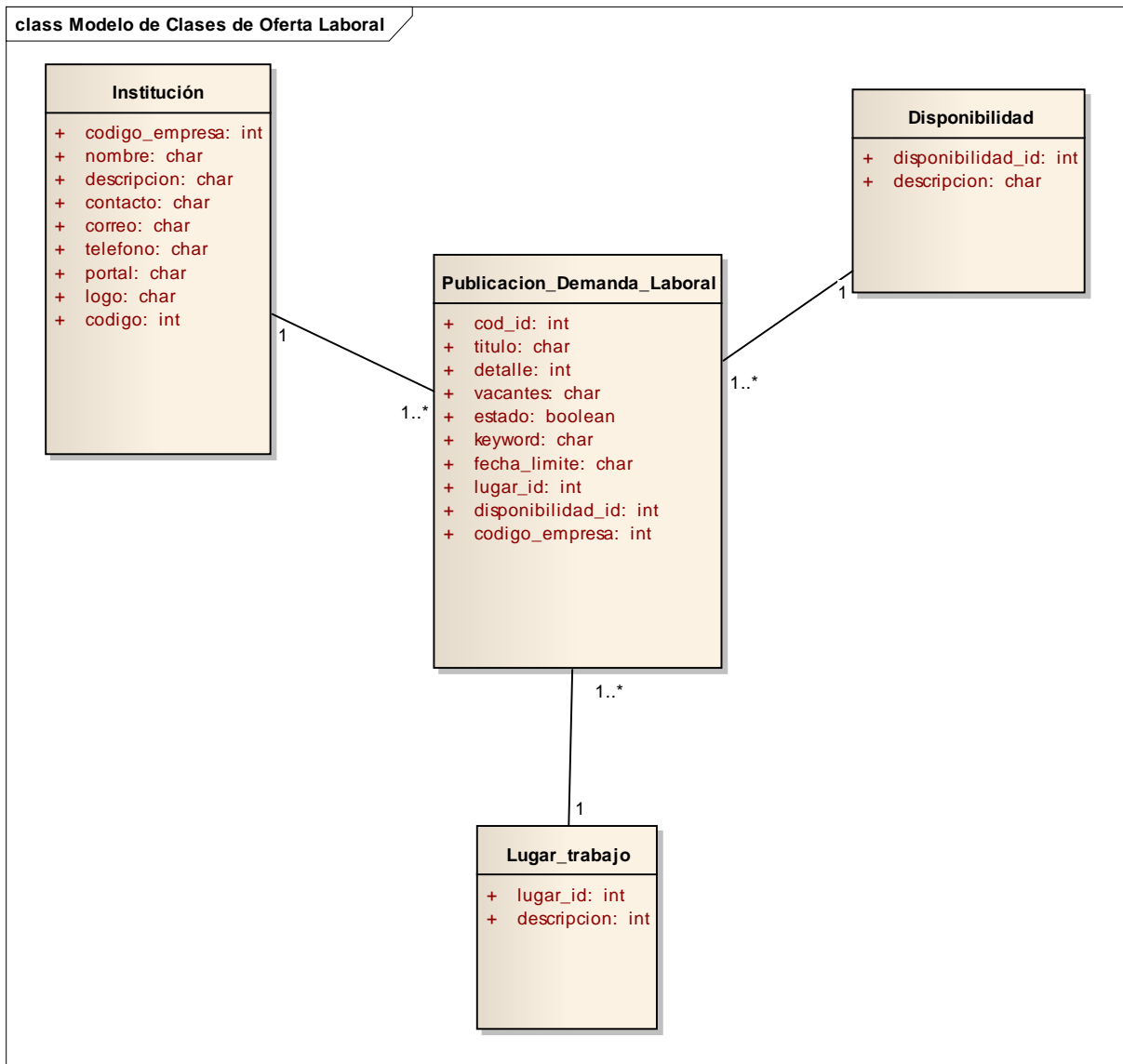


Figura 3. 6 Registro de Oferta Laboral

Tarea 2: Registro de datos de la demanda laboral.

La Figura 3.7, describe los campos que se registrara para la publicación de la oferta, relacionada con la institución.



Registrar Demanda Laboral	
Titulo	
Detalles	
Palabras claves	
Vacantes	1
Lugar de Trabajo	La Paz
Disponibilidad	Tiempo Completo
Guardar Limpiar	

Figura 3. 7 Pantalla de Registro de la Demanda Laboral

Pruebas de aceptación para la historia 2

a. Identificar todos los posibles resultados observables de la historia

- Interfaz para el llenado de datos de la demanda
- Pantalla para capturar datos de demanda laboral

b. Identificar los resultados que termina la historia y los que permiten continuar dentro de la historia.

- La historia termina cuando se termina de llenar los datos de la demanda laboral a registrar.

c. Identificar los caminos de ejecución posibles

- Comienza la historia cuando se registra datos de la demanda laboral.
- Valida los datos.
- La historia termina cuando completa los datos solicitados
- Almacena los datos en la base de datos.

d. Asigna un conjunto de valores válidos y valores del entorno a cada camino de ejecución para obtener el resultado esperado.

- Un conjunto de valores válidos está dado por el conjunto de datos compuesto por la información de la demanda laboral.

e. Eliminación de caminos redundantes

- No existen caminos redundantes

Historia de usuario 3: Búsqueda de Ofertas Laborales

Tarea 1: Búsqueda simple de Ofertas Laborales

La Figura 3.8 muestra la ventana de búsqueda rápida de ofertas laborales registradas.



Figura 3. 8 Pantalla de Búsqueda de Ofertas Laborales

Tarea 2: Búsqueda avanzada de Ofertas Laborales

En tanto que la Figura 3.9 muestra la ventana de búsqueda avanzada de ofertas laborales registradas.

Titulo	Detalle	Palabras Clave	Lugar	Ver Detalle
DBA SENIOR	DBA ORACLE	JAVA	Cochabamba	ver detalle
Programador Senior	SE necesita un programado Senior para el desarrollo de aplicaciones moviles	IOS APPLE	La Paz	ver detalle
PROGRAMADOR SENIOR	Se requiere un programado en JAVA con fuertes conocimientos en POD	JAVA	La Paz	ver detalle

Figura 3. 9 Pantalla de Búsqueda de Ofertas Laborales

Tarea 3: Mostrar detalle de la oferta laboral

La Figura 3.10 muestra la ventana del detalle de la oferta laboral.

Titulo	Detalle	Vacantes	Palabras Clave	Lugar	Disponibilidad	Fecha Limite
DBA SENIOR	DBA ORACLE	1	JAVA	Cochabamba	Tiempo Completo	12/12/2012

Institución

Logo	Detalle												
	<table border="1"><tr><td>Nombre</td><td>MINISTERIO DE EDUCACIÓN</td></tr><tr><td>Descripción</td><td>EMPRESA ESTATAL</td></tr><tr><td>Contacto</td><td>Juan Carlos Gutierrez</td></tr><tr><td>Correo</td><td>rock@rock.sap</td></tr><tr><td>Teléfono</td><td>234341</td></tr><tr><td>Portal</td><td>www.min-educacion.gob.bo</td></tr></table>	Nombre	MINISTERIO DE EDUCACIÓN	Descripción	EMPRESA ESTATAL	Contacto	Juan Carlos Gutierrez	Correo	rock@rock.sap	Teléfono	234341	Portal	www.min-educacion.gob.bo
Nombre	MINISTERIO DE EDUCACIÓN												
Descripción	EMPRESA ESTATAL												
Contacto	Juan Carlos Gutierrez												
Correo	rock@rock.sap												
Teléfono	234341												
Portal	www.min-educacion.gob.bo												

Numero de Publicaciones

Total	6 Publicaciones
-------	-----------------

Figura 3. 10 Pantalla para mostrar detalle de la oferta

Pruebas de aceptación para la historia 3

a. Identificar todos los posibles resultados observables de la historia

- Interfaz para el llenado para la búsqueda de ofertas laborales
- Pantalla para buscar ofertas laborales

b. Identificar los resultados que termina la historia y los que permiten continuar dentro de la historia.

- La historia termina cuando se termina de buscar las ofertas disponibles.

c. Identificar los caminos de ejecución posibles

- Comienza la historia cuando se escribe la búsqueda de alguna oferta laboral.
- Valida los datos.
- La historia termina cuando encuentra la oferta solicitada en la base de datos.
- Muestra el detalle de la oferta encontrada.

d. Asigna un conjunto de valores válidos y valores del entorno a cada camino de ejecución para obtener el resultado esperado.

- Un conjunto de valores válidos está dado por datos introducidos por el usuario.

e. Eliminación de caminos redundantes

- No existen caminos redundantes

Historia usuario 4: Modificación de Ofertas Laborales

Tarea 1: Edición de ofertas laborales

La Figura 3.11 muestra la ventana de edición de una oferta registrada.

Editar Oferta Cancelar [X]

Titulo	PROGRAMADOR JUNIOR	Palabras Clave	DESARROLLADOR	
Detalle ::	Nuestra empresa requiere programadores junior para proyecto		Vacantes	2
		Lugar de Trabajo	Santa Cruz	
		Disponibilidad	Medio Tiempo	

Actualizar Oferta

Figura 3. 11 Pantalla para editar Oferta Laboral

Tarea 2: Eliminación de ofertas laborales

La Figura 3.12 muestra la ventana de eliminación de una oferta laboral

Esta seguro de esta acción, el proceso es irreversible

Aceptar Cancelar

Titulo	Palabras Clave	Lugar	Acción
DEVELOPER JAVA	JAVA	La Paz	Editar Eliminar
PROGRAMADOR JUNIOR	DESARROLLADOR	Santa Cruz	Editar Eliminar
PROGRAMADOR SENIOR	JAVA	La Paz	Editar Eliminar

Figura 3. 12 Pantalla para la eliminación de una oferta laboral

Pruebas de aceptación para la historia 4

a. Identificar todos los posibles resultados observables de la historia

- Interfaz para la edición de ofertas laborales
- Pantalla para editar ofertas laborales

b. Identificar los resultados que termina la historia y los que permiten continuar dentro de la historia.

- La historia termina cuando se termina la actualización de los datos de la oferta laboral.

c. Identificar los caminos de ejecución posibles

- Comienza la historia cuando se escribe la búsqueda de alguna oferta laboral a editar o eliminarla.
- Visualiza los datos de la oferta a editar.
- La historia termina cuando se guarda los cambios pertinentes en la base de datos.
- La historia termina cuando se elimina la oferta laboral
- Muestra la oferta modificada o eliminada.

d. Asigna un conjunto de valores válidos y valores del entorno a cada camino de ejecución para obtener el resultado esperado.

- Un conjunto de valores válidos está dado por datos introducidos por el usuario.

e. Eliminación de caminos redundantes

- No existen caminos redundantes

3.8.3 TERCERA ITERACIÓN

En esta tercera iteración se realiza las tareas relacionadas con los postulantes.

Historia usuario 5: Registro de postulantes

Tarea 1: Diseño de una estructura de datos para guardar los datos de los postulantes (ver Figura 3.13).

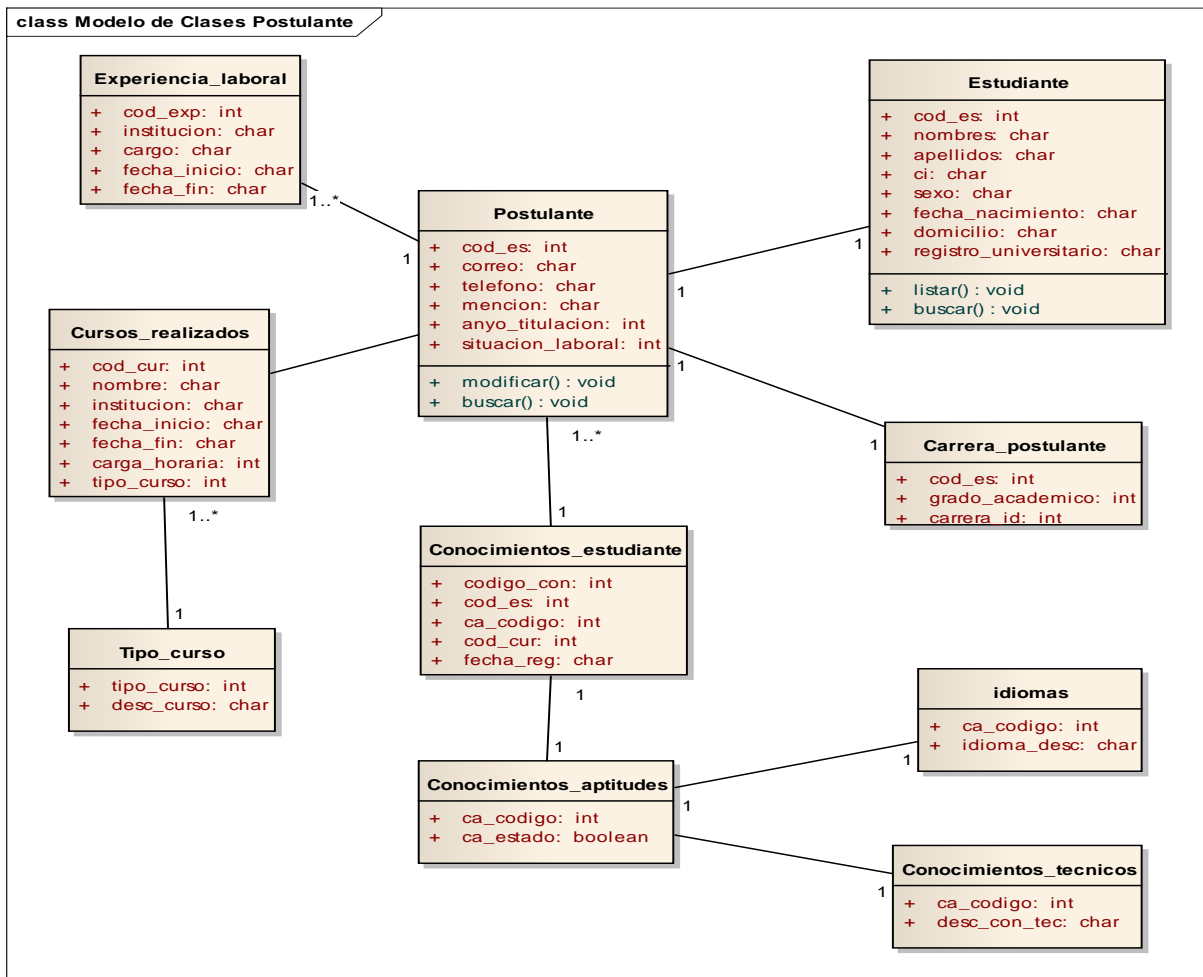


Figura 3. 13 Registro de Postulante

Tarea 2: Registro datos complementarios del postulante

La Figura 3.14 muestra la ventana de registro de datos personales del postulante

Datos Personales Conocimientos Experiencia Laboral Cursos Realizados			
Nombres	Apellidos	Fecha Nacimiento	Género
AMPARO	CALERNO ALIAGA	25/11/1981	FEMENINO
Completar Datos			
Correo Actual	Teléfono	Situación Laboral	
<input type="text"/>	<input type="text"/>	Disponible <input type="text" value="v"/>	
Grado Académico	Mención	Fecha Graduación	
LICENCIATURA <input type="text" value="v"/>	CIENCIAS DE LA COMPUTACION <input type="text" value="v"/>	<input type="text"/> <input type="text"/> <input type="text" value="v"/>	
<input type="button" value="Actualizar Datos"/>			

Figura 3. 14 Pantalla para capturar datos complementarios

Tarea 3: Registro de conocimientos o aptitudes del postulante

La Figura 3.15 muestra la ventana de registro de conocimientos del postulante

Datos Personales Conocimientos Experiencia Laboral Cursos Realizados
Sistemas Operativos
<input type="text"/>
Lenguajes de Programación
<input type="text"/>
Gestor de Base de Datos
<input type="text"/>
Frameworks
<input type="text"/>
CMS (Sistema de Gestión de Contenidos)
<input type="text"/>
Idiomas
<input type="text"/>
<input type="button" value="Actualizar Datos"/>

Figura 3. 15 Pantalla para capturar conocimientos o aptitudes del postulante

Tarea 3: Registro de experiencia laboral

La Figura 3.16 muestra la ventana de registro de experiencia laboral del postulante

The screenshot shows a web interface with a navigation bar at the top containing the following tabs: *Datos Personales*, *Conocimientos*, **Experiencia Laboral**, and *Cursos Realizados*. The main form area is titled 'Experiencia Laboral' and contains a table with the following structure:

Cargo	Institución	Fecha Inicio	Fecha Fin	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Guardar"/>

Below the table is a summary row with the following columns: **Cargo**, **Institución**, **Fecha Inicio**, **Fecha Fin**, and **Acción**. At the bottom of the form is a button labeled 'Actualizar Datos'.

Figura 3. 16 Pantalla para registrar la experiencia laboral del postulante

Tarea 4: Registro cursos realizados

La Figura 3.17 muestra la ventana de registro de cursos realizados por el postulante

The screenshot shows a web interface with a navigation bar at the top containing the following tabs: *Datos Personales*, *Conocimientos*, *Experiencia Laboral*, and **Cursos Realizados**. The main form area is titled 'Cursos Realizados' and contains a table with the following structure:

Tipo de Curso	Nombre	Institución	Fecha Inicio	Fecha Fin	Horas Académicas	Tipo Certificación	
<input type="text" value="Posgrado"/>	<input type="text" value="Nombre del Curso"/>	<input type="text" value="Nombre de la institución"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="APROBACIÓN"/>	<input type="button" value="Guardar"/>

Below the table is a summary row with the following columns: **Tipo de Curso**, **Nombre del Curso**, **Institución**, **Fecha Inicio**, **Fecha Fin**, **Horas Académicas**, **Tipo Certificación**, and **Acción**. At the bottom of the form is a button labeled 'Actualizar Datos'.

Figura 3. 17 Pantalla para registrar los cursos realizados del postulante

Pruebas de aceptación para la historia 5

a. Identificar todos los posibles resultados observables de la historia

- Interfaz para registro de datos personales
- Interfaz para registro de conocimientos
- Interfaz para registro de experiencia laboral
- Interfaz para registro de cursos realizados
- Pantalla para registrar datos de postulante

b. Identificar los resultados que termina la historia y los que permiten continuar dentro de la historia.

- La historia termina cuando se termina de completar los datos del postulante.

c. Identificar los caminos de ejecución posibles

- Comienza la historia cuando registra los datos del postulante como ser: datos personales, conocimientos, experiencia laboral y cursos realizados.
- La historia termina cuando se guarda los cambios pertinentes en la base de datos.
- La historia termina cuando se muestran los datos registrados

d. Asigna un conjunto de valores válidos y valores del entorno a cada camino de ejecución para obtener el resultado esperado.

- Un conjunto de valores válidos está dado por datos introducidos por el usuario.

e. Eliminación de caminos redundantes

- No existen caminos redundantes

Historia usuario 6: Búsqueda de postulantes

Tarea 1: Búsqueda simple de Postulantes

La Figura 3.18 muestra la búsqueda simple de postulantes



The screenshot shows a search interface with two tabs: 'Búsqueda rápida' (selected) and 'Búsqueda Avanzada'. Below the tabs is a search bar with the text 'codeigniter' and a placeholder text 'Buscar Postulante por conocimientos'. Below the search bar is a list of search results:

Nombre(s)	Apellido(s)	Correo	Grado Académico	Estado	Ver
LESLY	MOLINA FLORES	correo@net.soy	LICENCIATURA	Disponible	Detalle

Figura 3. 18 Ventana de búsqueda rápida

Tarea 2: Búsqueda avanzada de Postulantes

La Figura 3.19 muestra la búsqueda avanzada de postulantes registrados



The screenshot shows an advanced search interface with two tabs: 'Búsqueda rápida' and 'Búsqueda Avanzada' (selected). Below the tabs are several filter fields:

- Grado Académico: ESTUDIANTE
- Idiomas: Inglés
- Género: FEMENINO
- Experiencia Laboral: 1 Año
- Conocimientos: Java
- Cursos Realizados: Por ejemplo Oracle

Below the filters are 'Buscar' and 'Limpiar' buttons. Below the buttons is a list of search results:

Nombre(s)	Apellido(s)	Correo	Grado Académico	Estado	Ver
AMPARO	CALERNO ALIAGA	correo@amparo.es	ESTUDIANTE	Disponible	Detalle

Figura 3. 19 Ventana de búsqueda avanzada

Tarea 3: Detalle del Postulante

La Figura 3.20 muestra el detalle del postulante encontrado

Información del Postulante Cerrar [X]

Datos Personales

Nombre(s)	Apellido(s)	Género	Teléfono	Email	Grado Académico	Año de Titulación
AMPARO	CALERNO ALMAGA	FEMENINO	12345678	correo@amparo.es	ESTUDIANTE	NO DISPONIBLE

Experiencia Laboral

Institución	Cargo	Fecha Inicio	Fecha Fin	Experiencia
ENTEL S.A.	PROGRAMADOR JUNIOR	04/07/2011	09/11/2012	1 Año, 4 Meses y 5 Días
Experiencia Total				1 Año, 4 Meses y 5 Días

Cursos Realizados

Tipo	Nombre del Curso	Institución	Estado	Carga Horaria
Conferencia	SEGURIDAD INFORMÁTICA	UNIVERSIDAD NUR	APROBACIÓN	50 Horas

Conocimientos - Aptitudes

Sistemas Operativos	Android Mac OSX Windows 8
Lenguajes de Programación	Java Php Visual Basic .NET
Gestor de Base de Datos	PostgreSQL
Idiomas	Inglés Francés

Figura 3. 20 Ventana detalle del Postulante

Pruebas de aceptación para la historia 6

a. Identificar todos los posibles resultados observables de la historia

- Interfaz para la búsqueda rápida del postulante
- Interfaz para la búsqueda avanzada del postulante
- Interfaz para la información en detalle del postulante
- Pantalla para la búsqueda de postulante

b. Identificar los resultados que termina la historia y los que permiten continuar dentro de la historia.

- La historia termina cuando se termina de completar los datos de la búsqueda del postulante.

c. Identificar los caminos de ejecución posibles

- Comienza la historia cuando se escriben los datos que se buscan de un postulante como ser: conocimientos, cursos, etc.
- La historia termina cuando muestra la información detallada del postulante encontrado en la base de datos.

d. Asigna un conjunto de valores válidos y valores del entorno a cada camino de ejecución para obtener el resultado esperado.

- Un conjunto de valores válidos está dado por datos introducidos por el usuario.

e. Eliminación de caminos redundantes

- No existen caminos redundantes

Historia usuario 7: Edición de datos Postulante

Tarea 1: Actualización de datos complementarios del postulante

La Figura 3.21 muestra la ventana de actualización de datos personales del postulante

Datos Personales				Conocimientos	Experiencia Laboral	Cursos Realizados
Nombres		Apellidos		Fecha Nacimiento		Género
AMPARO		CALERNO ALIAGA		25/11/1981		FEMENINO
Actualizar Datos						
Correo Actual			Teléfono		Situación Laboral	
correo@amparo.es			12345678		Disponible	
Grado Académico			Mención		Fecha Graduación	
ESTUDIANTE			CIENCIAS DE LA COMPUTACION			
Actualizar Datos						

Figura 3. 21 Pantalla para actualizar datos complementarios

Tarea 2: Actualización de conocimientos o aptitudes del postulante

La Figura 3.22 muestra la ventana de actualización de conocimientos del postulante

The screenshot shows a web interface for updating skills. At the top, there are four tabs: 'Datos Personales', 'Conocimientos', 'Experiencia Laboral', and 'Cursos Realizados'. The 'Conocimientos' tab is active. Below the tabs, there are several sections, each with a title and a list of skills in a horizontal box with 'x' icons to remove them:

- Sistemas Operativos:** Android, Mac OSX, Windows B
- Lenguajes de Programación:** Java, Php, Visual Basic .NET
- Gestor de Base de Datos:** PostgreSQL
- Frameworks:** (empty)
- CMS (Sistema de Gestión de Contenidos):** (empty)
- Idiomas:** Ingles, Frances

At the bottom of the form is a button labeled 'Actualizar Datos'.

Figura 3. 22 Pantalla para capturar conocimientos o aptitudes del postulante

Tarea 3: Actualización de experiencia laboral

La Figura 3.23 muestra la ventana de actualización de la experiencia laboral del postulante

The screenshot shows a web interface for updating work experience. At the top, there are four tabs: 'Datos Personales', 'Conocimientos', 'Experiencia Laboral', and 'Cursos Realizados'. The 'Experiencia Laboral' tab is active. Below the tabs, there is a form with a table and a 'Guardar' button.

Cargo	Institución	Fecha Inicio	Fecha Fin	
				Guardar

Cargo	Institución	Fecha Inicio	Fecha Fin	Acción
PROGRAMADOR JUNIOR	ENTEL S.A.	04/07/2011	09/11/2012	Eliminar
PROGRAMADOR SENIOR	S.E.G.I.P.	07/11/2011	05/11/2012	Eliminar

At the bottom of the form is a button labeled 'Actualizar Datos'.

Figura 3. 23 Pantalla para registrar la experiencia laboral del postulante

Tarea 4: Actualización de cursos realizados

La Figura 3.24 muestra la ventana de actualización de cursos realizados por el postulante

Tipo de Curso	Nombre	Institución
Posgrado	Nombre del Curso	Nombre de la institución

Fecha Inicio	Fecha Fin	Horas Académicas	Tipo Certificación
			APROBACIÓN

Tipo de Curso	Nombre del Curso	Institución	Fecha Inicio	Fecha Fin	Horas Académicas	Tipo Certificación	Acción
Conferencia	SEGURIDAD INFORMÁTICA	UNIVERSIDAD NUR	01/11/2011	30/11/2011	50	APROBACIÓN	Eliminar

Figura 3. 24 Pantalla para registrar los cursos realizados del postulante

Pruebas de aceptación para la historia 7

a. Identificar todos los posibles resultados observables de la historia

- Interfaz para actualizar datos personales
- Interfaz para actualizar conocimientos
- Interfaz para actualizar experiencia laboral
- Interfaz para actualizar cursos realizados
- Pantalla para actualizar datos de postulante

b. Identificar los resultados que termina la historia y los que permiten continuar dentro de la historia.

- La historia termina cuando se termina de actualizar los datos del postulante.

c. Identificar los caminos de ejecución posibles

- Comienza la historia cuando se actualiza los datos del postulante como ser: datos personales, conocimientos, experiencia laboral y cursos realizados.

- La historia termina cuando se guarda los cambios pertinentes en la base de datos.
- La historia termina cuando se muestra los datos registrados

d. Asigna un conjunto de valores válidos y valores del entorno a cada camino de ejecución para obtener el resultado esperado.

- Un conjunto de valores válidos está dado por datos introducidos por el usuario.

e. Eliminación de caminos redundantes

- No existen caminos redundantes

3.8.4 CUARTA ITERACIÓN

En esta iteración se realiza las tareas referentes a los reportes estadísticos y las herramientas para el administrador del sistema.

Historia usuario 8: Reportes estadísticos

Tarea 1: Reporte estadístico de postulantes con conocimientos

La Figura 3.25 muestra los reportes estadísticos según los conocimientos de los postulantes registrados

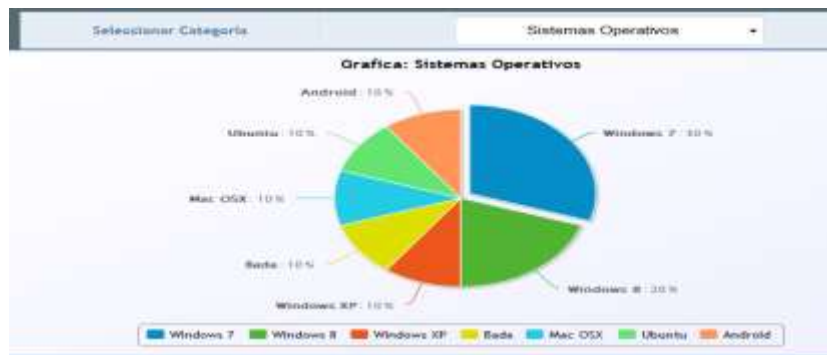


Figura 3. 25 Ventana que muestra datos estadísticos

Pruebas de aceptación para la historia 8

a. Identificar todos los posibles resultados observables de la historia

- Interfaz para visualizar los datos estadísticos
- Pantalla para visualizar los datos estadísticos de los postulantes referente a los conocimientos

b. Identificar los resultados que termina la historia y los que permiten continuar dentro de la historia.

- La historia termina cuando se termina de visualizar los datos estadísticos.

c. Identificar los caminos de ejecución posibles

- Comienza la historia cuando se elige una categoría de conocimientos.
- La historia termina cuando visualizar los datos estadísticos.

d. Asigna un conjunto de valores válidos y valores del entorno a cada camino de ejecución para obtener el resultado esperado.

- Un conjunto de valores válidos está dado por datos introducidos por el usuario.

e. Eliminación de caminos redundantes

- No existen caminos redundantes

Historia usuario 9: Privilegios de usuarios

Tarea 1: Administrar privilegios de usuarios

La Figura 3.26 muestra la búsqueda de usuarios en el sistema



Figura 3. 26 Muestra los usuarios registrados

La Figura 3.27 muestra la alta o baja de los usuarios

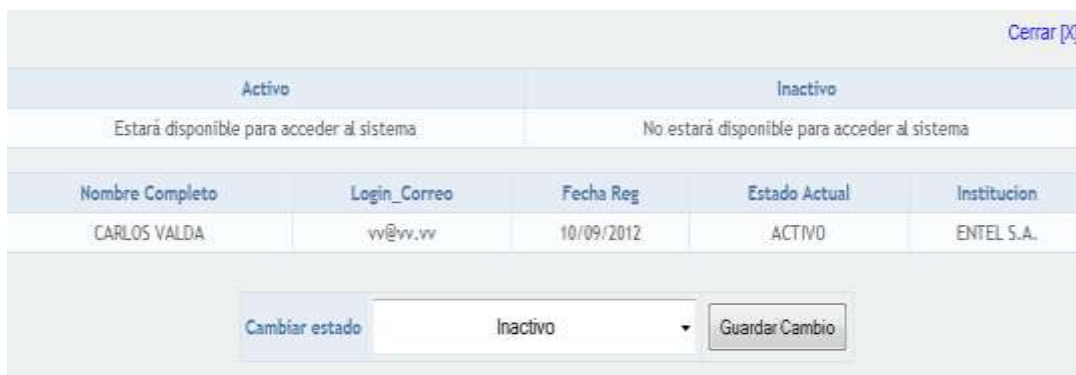


Figura 3. 27 Modificación de acceso al sistema

Tarea 2: Cambio de contraseña de usuarios

La Figura 3.28 muestra el cambio de contraseña por parte del administrador hacia los usuarios registrados en el sistema.



Figura 3. 28 Cambio de contraseña a los usuarios

Pruebas de aceptación para la historia 9

a. Identificar todos los posibles resultados observables de la historia

- Interfaz para buscar a los usuarios registrados en el sistema
- Interfaz para mostrar las búsquedas encontradas
- Interfaz para cambiar el estado del usuario
- Interfaz para cambiar la contraseña del usuario
- Pantalla para visualizar los cambios realizados sobre los usuarios en el sistema.

b. Identificar los resultados que termina la historia y los que permiten continuar dentro de la historia.

- La historia termina cuando se cambia el estado del usuario en la base de datos.
- La historia termina cuando se cambia la contraseña por parte del administrador y se muestra el cambio realizado.

c. Identificar los caminos de ejecución posibles

- Comienza la historia cuando se elige que tarea realizar, ya sea cambiar el estado del usuario o su contraseña.
- La historia termina cuando visualizar los cambios realizados en la base de datos sobre los usuarios.

d. Asigna un conjunto de valores válidos y valores del entorno a cada camino de ejecución para obtener el resultado esperado.

- Un conjunto de valores válidos está dado por datos introducidos por el usuario.

e. Eliminación de caminos redundantes

- No existen caminos redundantes

3.9 DIAGRAMA DE DESPLIEGUE

El siguiente diagrama describe el Despliegue del Sistema (ver Figura 3.29).

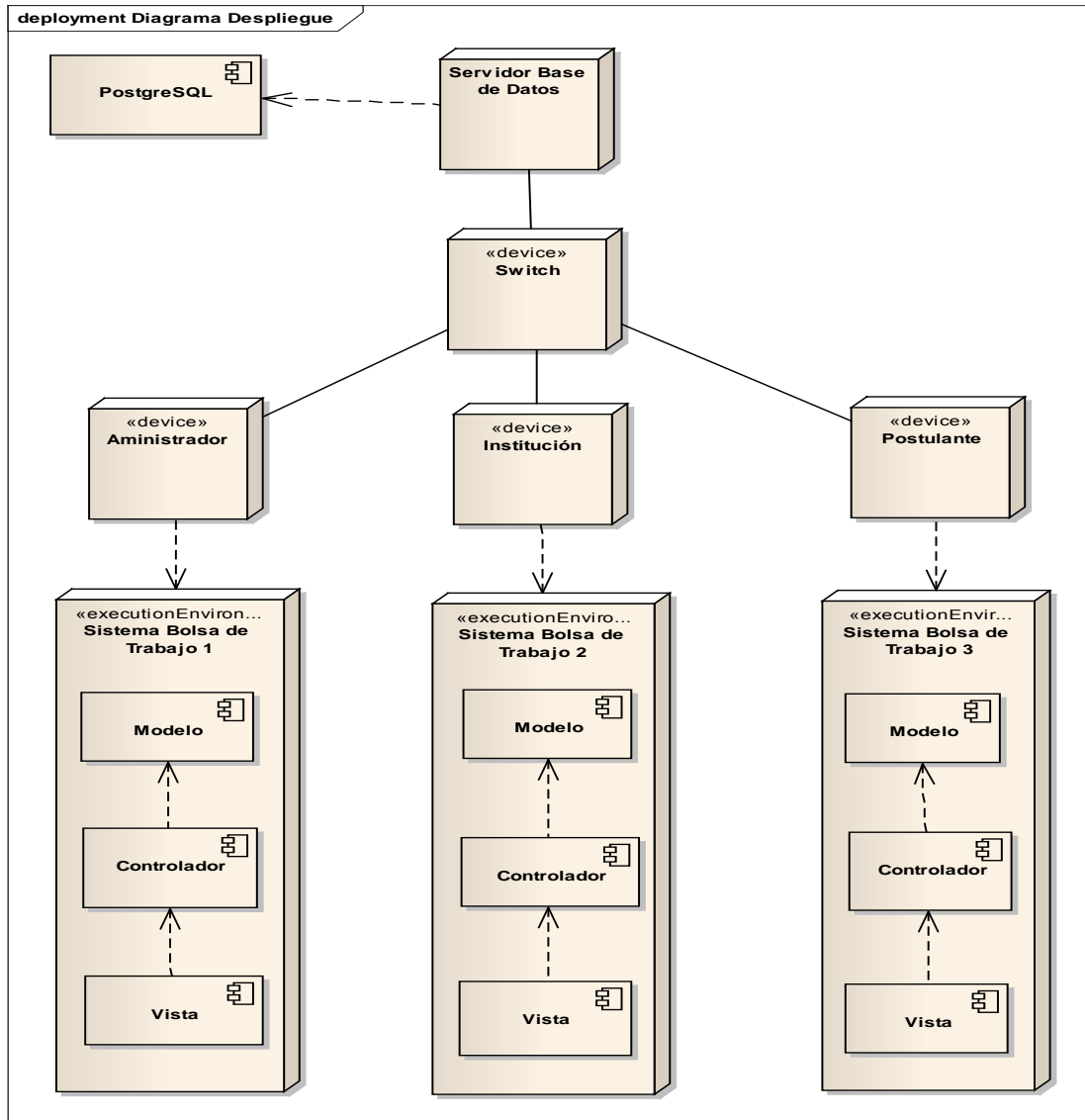


Figura 3. 29 Diagrama de Despliegue del sistema

CAPÍTULO IV

MÉTRICAS DE CALIDAD

4.1 INTRODUCCIÓN

Las métricas de calidad que serán aplicadas son la norma ISO-9126, y también se tomarán en cuenta las métricas de seguridad.

4.2 FUNCIONALIDAD

La funcionabilidad se midió a partir de las necesidades o requerimientos planteados, que se fueron cumpliendo a medida que se terminaba.

Los requerimientos funcionales son:

- Registro de Instituciones
- Registro de la Demanda
- Búsqueda de ofertas laborales
- Registro de Postulantes
- Búsqueda de Postulantes
- Herramientas administrativas

Adicionalmente se agrego

- Gráficos estadísticos de la demanda
- Tendencias del mercado Internacional

4.3 CONFIABILIDAD

El cálculo de la confiabilidad se lo realizará mediante el tiempo medio entre fallos (TMEF), usando la siguiente fórmula:

$$\text{TMEF} = \text{TMDf} + \text{TMDR}$$

Donde, a partir del desarrollo se obtuvieron los siguientes datos:

T MDF = 10 h

T MDR = 1 h

Con esto tenemos que:

TMEF = 10 h + 1 h

TMEF = 11 h

Ahora, la disponibilidad será calculada con la siguiente fórmula:

Disponibilidad = $[TMDf / (TMDf + TMDR)] \times 100 \%$

Disponibilidad = $[10 \text{ h} / (10 \text{ h} + 1 \text{ h})] \times 100 \%$

Disponibilidad = 90.91 %

Con esto se tiene que en un lapso de 11 horas existe un valor de 90.91 % de éxito de funcionamiento del sistema. Esto significa que el Sistema es confiable en un 90.91%

4.4 USABILIDAD

A partir de las pruebas realizadas con los usuarios, se apreció que:

- El Sistema posee fácil comprensión por parte de los usuarios a través de sus interfaces intuitivas.
- El Sistema posee facilidad de aprendizaje por parte de los usuarios gracias al uso de íconos descriptivos y de mensajes de ayuda en cada interfaz.

El Sistema es operable por parte de los usuarios ya que su uso es sencillo y dedicado a su objetivo.

4.5 EFICIENCIA

El Sistema es eficiente, debido a los siguientes parámetros:

- El tiempo de respuesta a cada petición.
- El tiempo mientras se realiza la captura, es utilizado por los usuarios para ir realizando la transcripción de los datos primarios.

Gracias a estos dos factores, el proceso de digitalización y transcripción se lo realiza en uno solo y los resultados son mejores que realizando los dos procesos por separado.

Los recursos utilizados por el Sistema son mínimos, gracias a la utilización de comandos de consola de Linux, los cuales ayudan bastante en la rapidez y el menor consumo de recursos.

4.6 FACILIDAD DE MANTENIMIENTO

El mantenimiento que pueda realizarse al Sistema no genera muchos costos, debido a que el código se encuentra con una arquitectura MVC³ y la modificación de alguna de estas partes es bastante sencilla.

4.7 PORTABILIDAD

El Sistema posee facilidad de ajuste, porque al haber sido realizado en Php, un lenguaje multiplataforma, entonces el Sistema puede funcionar bajo cualquier plataforma.

El Sistema posee facilidad de adaptación al cambio, porque al estar hecho basado en una Arquitectura Modelo – Vista – Controlador, la realización de algún cambio es sencilla y no genera problemas posteriores

4.8 INTEROPERABILIDAD

El sistema tiene una integración con el Sistema Integrado Académico (SIA), para el caso particular de los postulantes, ya que no se duplicara la información y mantendrá un acoplamiento homogéneo. Esto porque el SIA, es el vigente sistema, que maneja el historial académico de los estudiantes.

4.9 SEGURIDAD

El Sistema posee varios controles de seguridad:

³ Modelo Vista Controlador

A nivel de Sistema:

- Autenticación de usuarios mediante nombre de usuario y contraseña.
- Identificación de atribuciones según el tipo de usuario que introduzca su nombre de usuario y su contraseña.
- Verificación de usuarios conectados, evitando la conexión del mismo usuario desde dos puntos diferentes.
- Verificación de actividad o inactividad de cuentas en el momento de la conexión.
- Módulo de verificación de usuarios conectados.

A nivel de Base de Datos:

- Cambio de contraseñas de usuarios y roles de la Base de Datos una vez al mes.
- Manejo de roles a nivel de Base de Datos.
- Almacenamiento en una Base de Datos, desde varios clientes.

A nivel de Sistema Operativo:

- Uso de permisos de Linux en los archivos del Sistema Operativo.
- Cambio de contraseñas de usuarios del Sistema Operativo una vez al mes.

A nivel de Servidores:

- Acceso físico y remoto restringido a los Servidores de Base de Datos. Sólo el Área de Base de Datos tiene acceso físico y remoto a ellos.

Con todos estos medios de control de seguridad, el sistema posee bastantes niveles que lo ayudan a ser un sistema seguro.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Una vez concluido el desarrollo del proyecto, y logrado un sistema de información que cumple los requerimientos planteados por los clientes, acompañado de las pruebas de estrés, se puede asegurar el funcionamiento del sistema.

En cuanto a los objetivos específicos, se llegó a cumplir en su totalidad, detallando de la siguiente manera:

Se realizó una interfaz amigable de fácil acceso a los diferentes módulos que contiene el sistema.

Se hizo énfasis en las búsquedas de las ofertas laborales, como la búsqueda de los postulantes, de manera muy robusta.

Se construyó una base de datos, que se integró al sistema académico de la carrera de Informática (SIA), para trabajar de manera conjunta.

El proyecto ha sido realizado exitosamente bajo las especificaciones planteadas, en cuanto a la metodología que se usó para el desarrollo (Programación Extrema), que fue acorde a lo planificado, ya que las historias de usuario ayudaron a levantar la información de los requerimientos pero estuvo acompañado de la documentación de la mano de UML.

Existen debilidades del método empleado (Programación Extrema), ya que no va de la mano de la documentación formal. En un futuro sería mejor fusionar algunos métodos para sacar un híbrido que ayude a desenvolverse mejor.

5.2 RECOMENDACIONES

Las recomendaciones que se plantean son las siguientes:

Se recomienda realizar el mantenimiento preventivo y correctivo de los equipos donde se alojan las aplicaciones, como las bases de datos.

Se recomienda tener políticas de seguridad para el cambio de contraseña, a nivel de aplicación, base de datos y sistemas operativos.

Se recomienda que se realice auditorias web, para controlar las vulnerabilidades que pueden existir dentro de los sistemas que están publicados.

Se recomienda que no se abandone el sistema, que tenga un tiempo de vida útil para los fines con lo que fue creado.

Se recomienda promocionar el sistema de Bolsa de Trabajo, mediante redes sociales, para una mayor expansión y difusión.

Existen restricciones con respecto a las instituciones, en el caso del registro, ya que tiene que pasar por una verificación por parte del administrador, esto es debido a que se evite la trate y tráfico de personas, al momento de publicar las ofertas y demandas

BIBLIOGRAFÍA

- Beck, K. (2000). *Extreme programming explained*. Addison - Wesley.
- Calderón Rodríguez, M. A. (2004). *Sistema de Información cliente/servidor para los procesos de la bolsa de trabajo*. La Paz, Murillo, Bolivia: Universidad Mayor de San Andres.
- Fowler, M., & Beck, K. (2000). *Planning Extreme Programming*. Addison - Wesley.
- Gómez Garduza, G. B. (2010). *Sistema de Información para Bolsa de Trabajo*. Veracruz, México: Universidad Veracruzana - Facultad de Contaduría y Administración.
- Gutierrez Salas, M. A. (2004). *Sistema Automatizado de la Bolsa de Trabajo Decente*. La Paz, Muriilo, Bolivia: Universidad Mayor de San Andres.
- Hurtado, J. (2004). *Calidad de Ingeniería de Software - Programación Extrema como Proceso de Aseguramiento de la calidad en proyectos Web*.
- ISO-9126. (1947, Febrero 27). *ISO - International Organization for Standardization*. Retrieved 11 10, 2012, from <http://www.iso.org/iso/home.html>
- Larman, C. (2004). *UML y patrones*. Mexico: Prentice Hall.
- Muller, P.-A. (2000). *Modélisation object avec UML*. Paris: Eyrolles.
- OIT. (1919). *Organización Internacional del Trabajo*. Retrieved 11 12, 2012, from <http://www.ilo.org/>
- Parodi Ledezma, J., & Ruiz Jaramillo, I. (2005). *Proyecto E-Work - Diseño de un Sistema de Información Facilitador en la Colocación de Estudiantes y/o Profesionales del Instituto de Ciencias Humanísticas y Económicas en el Sector Empresarial, Utilizando Tecnología Internet*. Quito, Ecuador: Escuela Superior Politécnica del Litoral.
- Pressman, R. (2010). *Ingeniería de Software - Un Enfoque Practico*. Mc Graw-hill.
- Quinceno, F. (2011, Abril 12). *analisisydeseoiii.blogspot.com*. Retrieved Octubre 18, 2012, from analisisydeseoiii.blogspot.com: analisisydeseoiii.blogspot.com
- Sauceda Gastélum, G. (2008, junio). *Diseño de una Bolsa de Trabajo en una Unviersidad*. Retrieved 10 11, 2012, from http://www.itson.mx/publicaciones/pacioli/Documents/no63/4b-siseno_de_bolsa_de_trabajo_modificado_2.pdf
- Schmuller, J. (2002). *UML*. Mexico: Pearson.
- Stallman, R. (2004). *Software libre para una sociedad libre*. Madrid: Traficantes de Sueños c\hortaleza 19, 1º drcha.
- Wells, D. (2009). *Extremme Programaing*. Retrieved Octubre 08, 2012, from *Extremme Programaing*: www.extremeprogramming.org

ANEXO A: MANUAL DE USUARIO

Pantalla Principal



En esta ventana se muestra la pantalla principal del sistema Bolsa de Trabajo, donde se muestra el menú de opciones que se tiene.

Demanda Laboral

En este enlace accede a un submenú donde se puede buscar las ofertas registradas por instituciones inscritas en el sistema.



Una vez dado click a Búsqueda de Oferta, se ve una ventana que tiene búsqueda rápida de ofertas y avanzada, ahora se muestra la búsqueda rápida a continuación

Titulo	Detalle	Palabras Clave	Lugar	Ver
PROGRAMADOR JUNIOR	desarrollador de programas informáticos	PROGRAMADOR JAVA	La Paz	Detalle
PROGRAMADOR JAVA	se requiere programador junior en java	JAVA	La Paz	Detalle
PROGRAMADOR JUNIOR	Nuestra empresa requiere programadores junior para proyecto	DESARROLLADOR	Santa Cruz	Detalle

También se tiene una búsqueda avanzada que se muestra a continuación

Título	Detalle	Palabras Clave	Lugar	Ver
PROGRAMADOR JUNIOR	desarrollador de programas informáticos	PROGRAMADOR JAVA	La Paz	Detalle
PROGRAMADOR JUNIOR	Nuestra empresa requiere programadores junior para proyecto	DESARROLLADOR	Santa Cruz	Detalle

En esta búsqueda tiene las opciones de encontrar una oferta laboral registrada de varias formas, entre las cuales está la búsqueda por los siguientes parámetros:

- **Título.** El título de la oferta
- **Detalle.** El detalle que se tiene contiene la oferta
- **Keyword.** Palabras claves que describen la oferta laboral
- **Institución.** Lista de las instituciones que están registradas
- **Lugar.** El lugar donde se requiere de personal (por ejemplo: La Paz, El Alto, etc)
- **Disponibilidad.** El tiempo que se requiere que trabaje (por ejemplo: Tiempo completo, medio tiempo, etc.)

Luego de la búsqueda, que se muestra en la ventana de resultados, se tiene un link donde se puede ver el detalle de la oferta encontrada.

Título	Detalle	Palabras Clave	Lugar	Ver
PROGRAMADOR JUNIOR	desarrollador de programas informáticos	PROGRAMADOR JAVA	La Paz	Detalle
PROGRAMADOR JUNIOR	Nuestra empresa requiere programadores junior para proyecto	DESARROLLADOR	Santa Cruz	Detalle

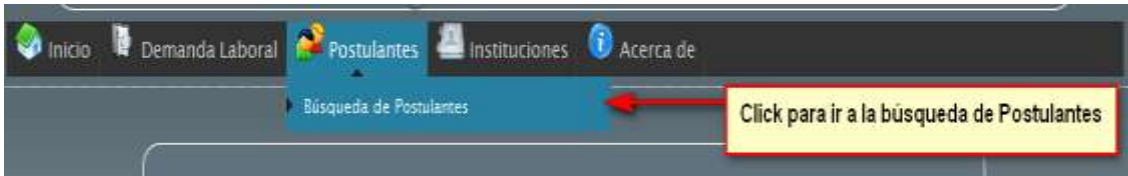
A continuación sale la siguiente ventana



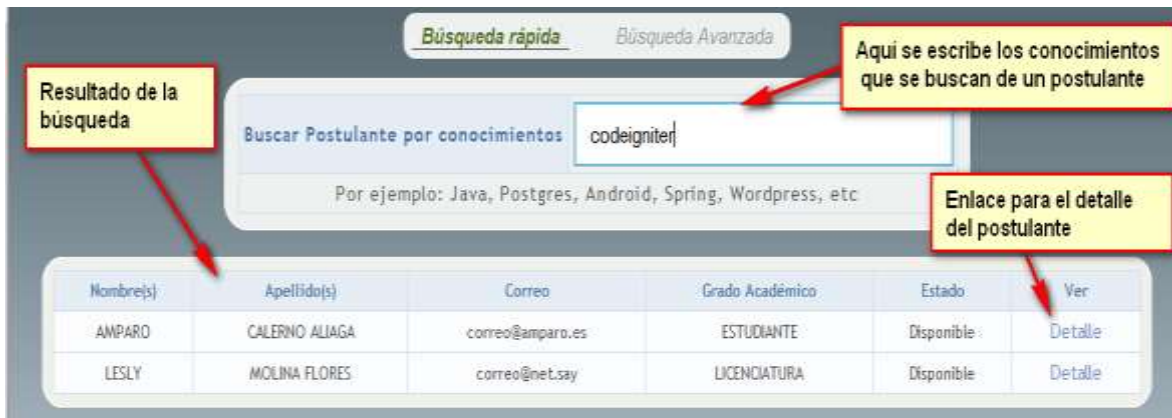
Muestra el detalle completo de la oferta laboral, donde se tiene además, la institución que publico la oferta, también está la información detallada de la misma para contactarse.

Postulantes

A continuación se tiene el enlace para mostrar la ventana de búsqueda de los postulantes.



Una vez hecho click sobre el enlace se muestra la ventana donde se tiene dos opciones para la búsqueda de los postulantes tanto simple como avanzada. A continuación se muestra la búsqueda rápida



En este tipo de búsquedas, basta con escribir en la caja de texto, para que las búsquedas se realicen de manera automática, los resultados dependerán de los conocimientos que se requieren de los postulantes, que cumplen los requisitos de la búsqueda que se realice

A continuación se muestra la búsqueda avanzada para los postulantes, que se muestran en la siguiente figura:

Nombre(s)	Apellido(s)	Correo	Grado Académico	Estado	Ver
AMPARO	CALERNO ALIAGA	correo@ampara.es	ESTUDIANTE	Disponible	Detalle

Este tipo de búsqueda, es más completo, considera varios aspectos que son muy interesantes, estos atributos de se describen a continuación:

- **Grado Académico.** En esta opción es para buscar por grado académico como ser: Estudiante, Egresado y Licenciado.
- **Género.** Esta opción describe si la búsqueda filtrar al género Masculino o Femenino.
- **Conocimientos.** En esta opción se puede escribir los conocimientos que se necesita de un postulante, como por ejemplo: Java, Php, Android, Postgres, Wordpress, etc. Puede colocar varios conocimientos (más de uno).
- **Idiomas.** Esta opción esta los idiomas disponibles como ser: Ingles, Frances, etc.
- **Experiencia Laboral.** Esta opción es para filtrar a los postulantes que tengan cierta cantidad de años de experiencia, la cual varía de 1 a 10 años.
- **Cursos Realizados.** Esta opción es para escribir algún curso que realizo el postulante, como por ejemplo: Oracle, Linux, Seguridad.

Luego de la búsqueda realizada se pude ir al detalle de la información del postulante que se describe en la siguiente imagen

Nombre(s)	Apellido(s)	Correo	Grado Académico	Estado	Ver
AMPARO	CALERNO ALIAGA	correo@amparo.es	ESTUDIANTE	Disponible	Detalle

Click para ver la información detallada del postulante

Luego se visualiza el detalle, que se muestra a continuación.

Información del Postulante Click o Esc para salir [Cerrar](#)

Datos Personales

Nombre(s)	Apellido(s)	Genero	Teléfono	Email	Grado Académico	Año de Titulación
AMPARO	CALERNO ALIAGA	FEMENINO	12345678	correo@amparo.es	ESTUDIANTE	NO DISPONIBLE

Experiencia Laboral

Institución	Cargo	Fecha Inicio	Fecha Fin	Experiencia
ENTEL S.A.	PROGRAMADOR JUNIOR	04/07/2011	09/11/2012	1 Año, 4 Meses y 5 Días

Experiencia Total: 1 Año, 4 Meses y 5 Días

Cursos Realizados

Tipo	Nombre del Curso	Institución	Estado	Carga Horaria
Conferencia	SEGURIDAD INFORMÁTICA	UNIVERSIDAD MUR	APROBACIÓN	30 Horas

Conocimientos - Aptitudes

Sistemas Operativos: Android, Mac OSX, Windows 8

Lenguajes de Programación: Java, Jsp, Visual Basic .NET

Gestor de Base de Datos: PostgreSQL

Idiomas: Inglés, Frances

Registro de Instituciones

En este apartado se muestra como registrar una Institución en el sistema, en el menú que hay se tiene que ir al siguiente enlace:



Una vez realizado esta acción sale la siguiente ventana:

El formulario está dividido en dos secciones: "Datos del Encargado" y "Datos de la Empresa".

- Datos del Encargado:** Campos para Nombre(s), Apellido(s), correo electrónico (ejemplo: usuario@ejemplo.com), Contraseña y Confirmar Contraseña.
- Datos de la Empresa:** Campos para Nombre Empresa, Descripción, correo electrónico (ejemplo: usuario@empresa.com), Teléfono y URL (ejemplo: http://www.institución.com).

Un botón "Registrar" está ubicado al final del formulario. Una etiqueta amarilla con el texto "Boton para Registrar la nueva Institución" tiene una flecha roja que apunta directamente al botón "Registrar".

Se registran los datos del encargado de la institución, como los datos de la misma.

Acceso al sistema.

El sistema cuenta con tres tipos de usuarios, como ser: Administrador, Instituciones y los Postulantes. Ahora se detallara como pueden acceder al sistema, de manera que se habiliten más opciones, para los usuarios que pueden loguearse al sistema.

En la siguiente pantalla muestra cómo pueden loguearse al sistema:



Después de dar click en el enlace sale esta ventana

La pantalla tiene el título "Iniciar Sesión" y dos campos de entrada: "usuario" y "Contraseña", con un botón "Entrar" a la derecha.

Un recuadro amarillo con el texto "Instrucciones para iniciar sesión en el sistema" tiene una flecha roja que apunta hacia abajo a un área de texto con las siguientes instrucciones:

- Si es estudiante de la Carrera de Informática, ingresan con los datos del sistema SIGA (Sistema de Gestión Académica) excepto la contraseña
- Ejemplo: usuario:ZZA251181 y la contraseña:CI (Carnet de Identidad)
- Si es otro usuario, se ingresa con los datos que le dio el sistema

Caso Administrador

Si el usuario es el administrador le saldrá el siguiente menú:



Listado de instituciones registradas



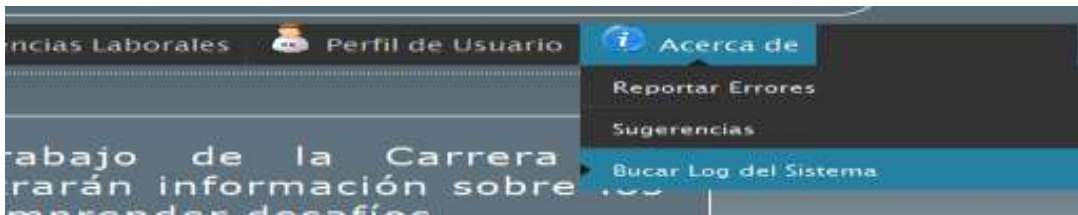
Una vez hecho esta acción se muestra las instituciones registradas

Nº	Logo	Nombre	Encargado	Estado	Cambiar Estado	Detalle de la Institución
1		ACOS LIMITADA	Carlos Ticona	Activo	Cambiar Estado	Detalle
2		ENTEL S.A.	CARLOS VALDA	Activo	Cambiar Estado	Detalle
3		MINISTERIO DE EDUCACIÓN	Andres Solíz	Activo	Cambiar Estado	Detalle
4		ELECTROPAZ	Isabel Fernandez	Activo	Cambiar Estado	Detalle
5		COBBE	Pamela Limachi	Activo	Cambiar Estado	Detalle
6		GOBIERNO MUNICIPAL AUTÓNOMO DE EL ALTO	Roger Altamirano Ibañez	Inactivo	Cambiar Estado	Detalle
7		MINISTERIO DE LA PRESIDENCIA	CARLOS VALDA MAMANI	Inactivo	Cambiar Estado	Detalle
8		MINISTERIO DE GOBIERNO	CARLOS MAMANI	Inactivo	Cambiar Estado	Detalle

Se puede visualizar el detalle de la institución, que se puede apreciar en la siguiente imagen.



El administrador cuenta con la vista de log del sistema que se puede apreciar en la siguiente imagen

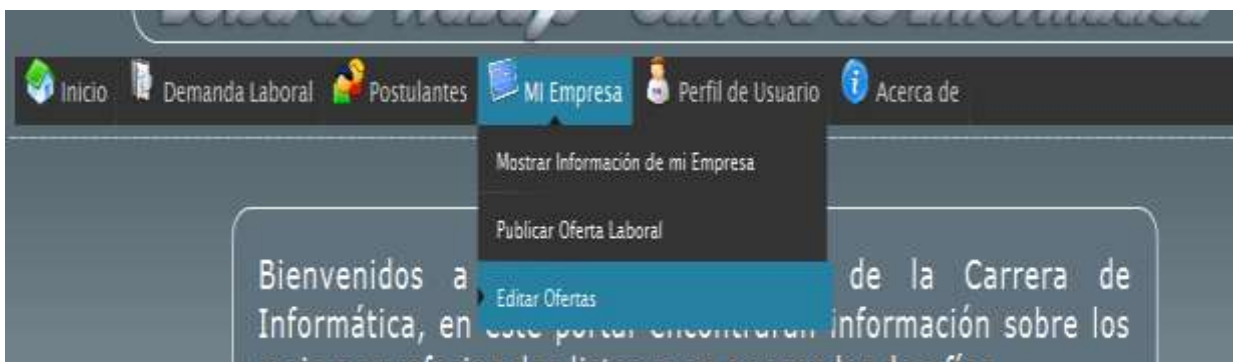


Luego de dar click sobre este enlace, muestra la siguiente ventana




Usuario Institución

Si el usuario es una institución mostrara el siguiente menú:



Dentro de este menú hay opciones como mostrar información de la institución como se muestra en la siguiente imagen:

Información detallada



Nombre	ENTEL S.A.
Descripción	ENTIDAD DEL GOBIERNO NACIONAL
Contacto	CARLOS VALDA GONZALES
Correo	contacto@entel.bo
Teléfono	2343453
Portal	www.entel.bo

También el registro de Ofertas Laborales

Registrar Demanda Laboral

Vacantes	1
Lugar de Trabajo	La Paz
Disponibilidad	Tiempo Completo

Y la edición de ofertas, donde se puede modificar una oferta y eliminar la oferta.

Busqueda de ofertas registradas

Editar Ofertas

Editar la oferta

Eliminar la oferta

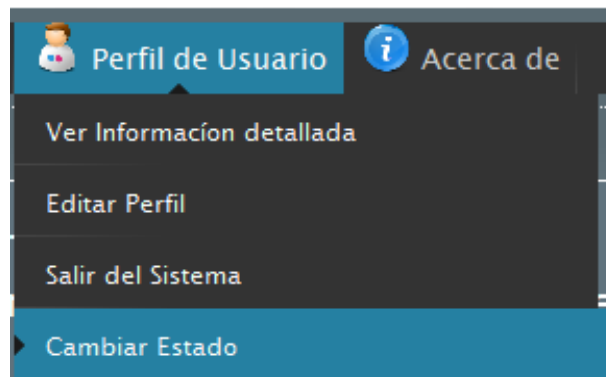
Titulo	Detalle	Palabras Clave	Lugar	Acción
PROGRAMADOR C#	DESARROLLADOR EN .NET, con base sólida en C, C++	C# C++	Santa Cruz	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Tanto el Administrador y la Institución cuentan con el cambio de contraseña en el sistema, como se muestra en la siguiente imagen



Usuario Postulante

A diferencia de los otros usuarios, tiene los siguientes menus



Donde se puede ver la información completa del postulante, como se aprecia en la siguiente imagen



También cuenta con la edición de sus datos

Datos Personales *Conocimientos* *Experiencia Laboral* *Cursos Realizados*

Nombres	Apellidos	Fecha Nacimiento	Género
AMPARO	CALERNO ALIAGA	25/11/1981	FEMENINO

Actualizar Datos

Correo Actual	Teléfono	Situación Laboral
borreo@amparo.es	12345678	Disponible
Grado Académico	Mención	Fecha Graduación
ESTUDIANTE	CIENCIAS DE LA COMPUTACION	

Actualizar Datos

Conocimientos y/o aptitudes

Datos Personales **Conocimientos** *Experiencia Laboral* *Cursos Realizados*

Sistemas Operativos

Android x Mac OSX x Windows 8 x

Lenguajes de Programación

Java x Php x Visual Basic .NET x

Gestor de Base de Datos

PostgreSQL x

Frameworks

CMS (Sistema de Gestión de Contenidos)

Idiomas

Inglés x Francés x

Experiencia Laboral

Datos Personales *Conocimientos* **Experiencia Laboral** *Cursos Realizados*

Cargo	Institución	Fecha Inicio	Fecha Fin	
				Guardar

Cargo	Institución	Fecha Inicio	Fecha Fin	Acción
PROGRAMADOR JUNIOR	ENTEL S.A.	04/07/2011	09/11/2012	Eliminar

Actualizar Datos

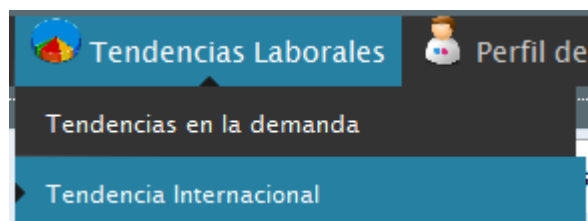
Y Cursos Realizados

Tipo de Curso	Nombre del Curso	Institución	Fecha Inicio	Fecha Fin	Horas Académicas	Tipo Certificación	Acción
Conferencia	SEGURIDAD INFORMÁTICA	UNIVERSIDAD NUR	01/11/2011	30/11/2011	50	APROBACIÓN	Eliminar

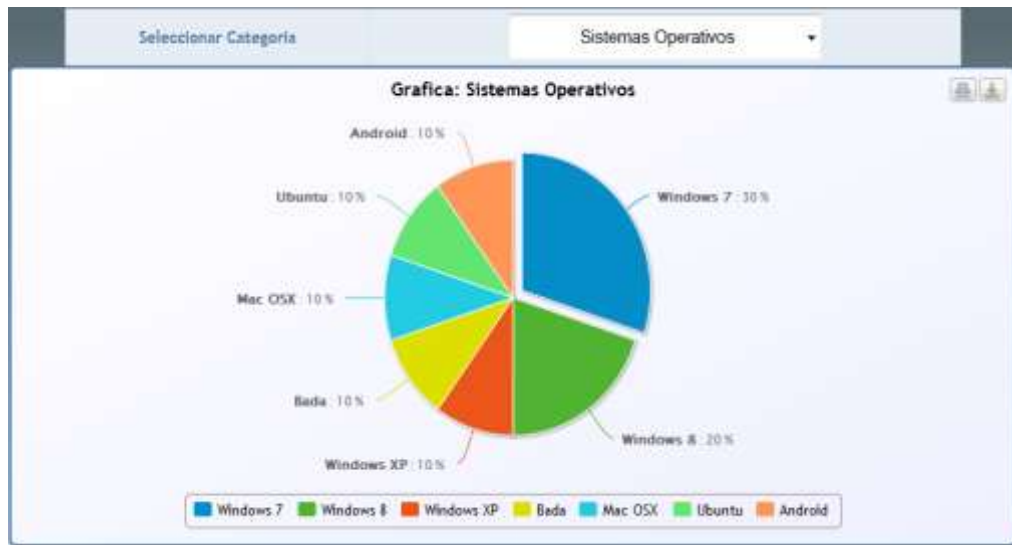
También cuenta con el cambio de estado, esto para salir o no en las búsquedas que se realicen por el sistema.

Activo	Inactivo
Estará disponible para las búsquedas del sistema	No estará disponible para las búsquedas del sistema

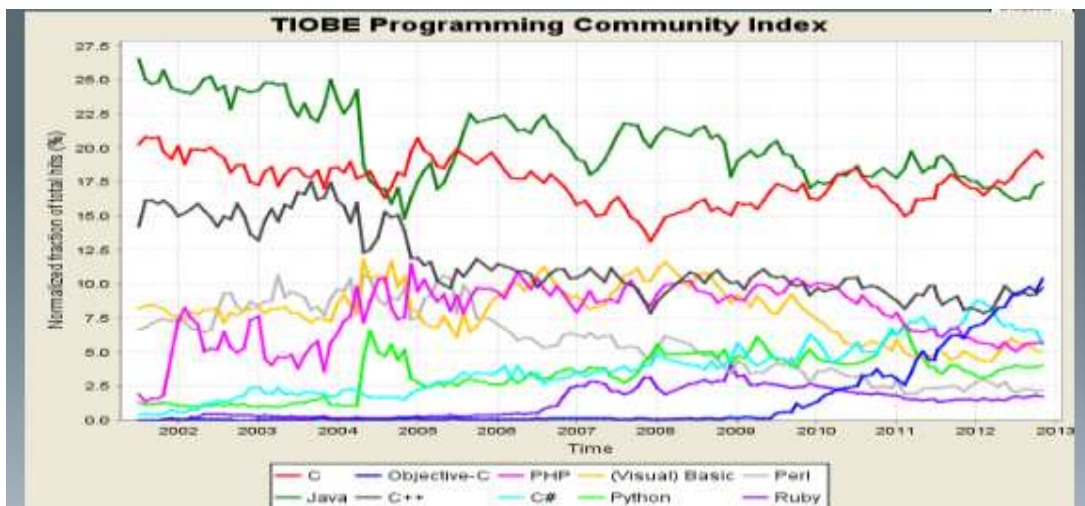
Es sistema cuenta con las estadísticas sobre los conocimientos, como se aprecia en la siguiente figura



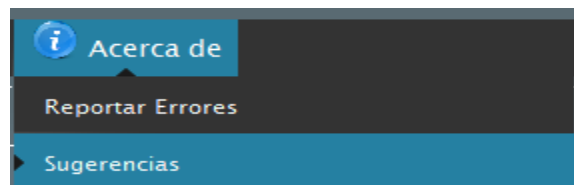
Tendencias de la demanda



Tendencia internacional



Reportes de Log's



Reporte de Errores



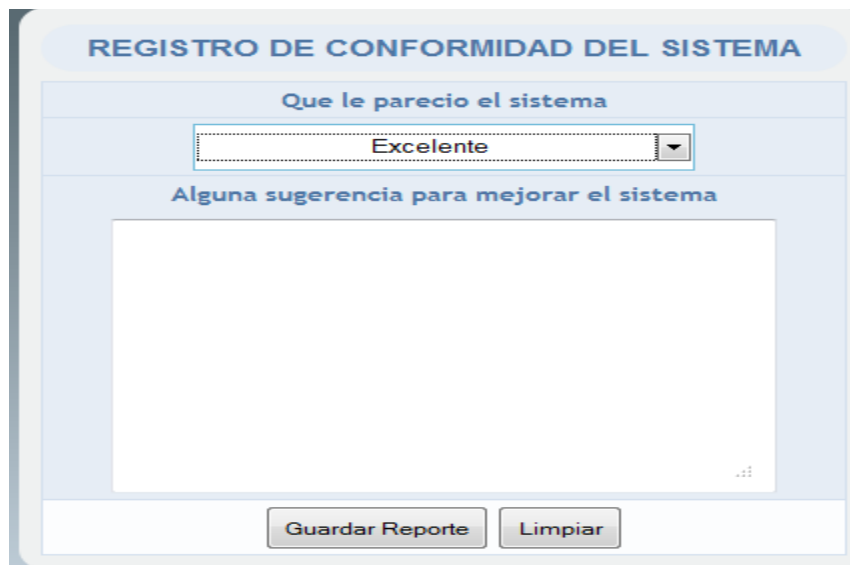
REGISTRO DE ERRORES DEL SISTEMA

Título del error o bug

Detalle del error

Guardar Reporte Limpiar

Sugerencias



REGISTRO DE CONFORMIDAD DEL SISTEMA

Que le pareció el sistema

Excelente

Alguna sugerencia para mejorar el sistema

Guardar Reporte Limpiar

ANEXO B: HISTORIAS DE USUARIO

Modulo Demanda Laboral

Historia de Usuario	
Numero: 2	Usuario: Institución
Nombre Historia: Registro de demanda laboral	
Prioridad del negocio: Alta	Riesgo en desarrollo: Alta (Alta/Media/Baja)
Puntos estimados: 1	Iteración asignada: 1
Descripción: Se registrara la demanda laboral de las instituciones.	
Observaciones: Previo registro de la institución.	

Tarea	
Numero: 2.1	Número historia: 2
Nombre Tarea: Diseñar una estructura de datos para registrar de la demanda laboral	
Tipo de tareas: Diseño	Puntos estimados: 1/2
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se realiza el diseño de la base de datos para el registro de la demanda laboral	

Tarea	
Numero: 2.2	Número historia: 2
Nombre Tarea: Registro de datos de la demanda laboral	
Tipo de tareas: Desarrollo	Puntos estimados: 1
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana en la cual se introducirá datos de la demanda laboral.	

Historia de Usuario	
Numero: 3	Usuario: Institución, Administrador y Postulante
Nombre Historia: Búsqueda de Ofertas Laborales	
Prioridad del negocio: Alta	Riesgo en desarrollo: Alta (Alta/Media/Baja)
Puntos estimados: 1	Iteración asignada: 2
Descripción: Se realizará la búsqueda de ofertas Laborales.	
Observaciones: Previo registro de la ofertas laborales.	

Tarea	
Numero: 3.1	Número historia: 3
Nombre Tarea: Búsqueda simple de ofertas laborales	
Tipo de tareas: Desarrollo	Puntos estimados: 1
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde se busca de forma simple o rápida las ofertas laborales registradas.	

Tarea	
Numero: 3.2	Número historia: 3
Nombre Tarea: Búsqueda avanzada de ofertas laborales	
Tipo de tareas: Desarrollo	Puntos estimados: 1/2
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde se busca de forma avanzada la oferta laboral.	

Tarea	
Numero: 3.3	Número historia: 3
Nombre Tarea: Mostrar detalle de oferta laboral	
Tipo de tareas: Desarrollo	Puntos estimados: 1/2
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde se muestre el detalle de la oferta laboral.	

Historia de Usuario	
Numero: 4	Usuario: Institución
Nombre Historia: Edición de Ofertas Laborales	
Prioridad del negocio: Media	Riesgo en desarrollo: Media (Alta/Media/Baja)
Puntos estimados: 1	Iteración asignada: 2
Descripción: Se realizará la búsqueda de ofertas Laborales.	
Observaciones: Previo registro de la ofertas laborales.	

Tarea	
Numero: 4.1	Número historia: 4
Nombre Tarea: Edición de ofertas laborales.	
Tipo de tareas: Desarrollo	Puntos estimados: 1
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde se edite las ofertas laborales registradas.	

Tarea	
Numero: 4.2	Número historia: 4
Nombre Tarea: Eliminación de ofertas laborales	
Tipo de tareas: Desarrollo	Puntos estimados: 1
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde se elimine las ofertas laborales registradas de una determinada institución.	

Modulo Postulantes

Historia de Usuario	
Numero: 5	Usuario: Postulante
Nombre Historia: Registro de Postulantes	
Prioridad del negocio: Alta	Riesgo en desarrollo: Alta (Alta/Media/Baja)
Puntos estimados: 2	Iteración asignada: 3
Descripción: Se registra lo datos de los postulantes	
Observaciones: Confirmado por el cliente.	

Tarea	
Numero: 5.1	Número historia: 5
Nombre Tarea: Diseño de una estructura de datos para guardar los datos de postulantes	
Tipo de tareas: Desarrollo	Puntos estimados: 1
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una estructura de datos para guardar los datos de los postulantes en la base de datos..	

Tarea	
Numero: 5.2	Número historia: 5
Nombre Tarea: Registro datos complementarios del postulante	
Tipo de tareas: Desarrollo	Puntos estimados: 1/2
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde se registre los datos complementarios del postulante.	

Tarea	
Numero: 5.3	Número historia: 5
Nombre Tarea: Registro de conocimientos o aptitudes del postulante	
Tipo de tareas: Desarrollo	Puntos estimados: 1/2
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde se registre los conocimientos o aptitudes del postulante.	

Tarea	
Numero: 5.4	Número historia: 5
Nombre Tarea: Registro de experiencia laboral	
Tipo de tareas: Desarrollo	Puntos estimados: 1/2
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde se registre la experiencia laboral del postulante.	

Tarea	
Numero: 5.5	Número historia: 5
Nombre Tarea: Registro cursos realizados	
Tipo de tareas: Desarrollo	Puntos estimados: 1/2
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde se registre los cursos realizados del postulante.	

Historia de Usuario	
Numero: 6	Usuario: Institución, Administrador y Postulante
Nombre Historia: Búsqueda de Postulantes	
Prioridad del negocio: Alta	Riesgo en desarrollo: Alta (Alta/Media/Baja)
Puntos estimados: 1	Iteración asignada: 3
Descripción: Se realizará la búsqueda de ofertas Laborales.	
Observaciones: Previo registro del postulante.	

Tarea	
Numero: 6.1	Número historia: 6
Nombre Tarea: Búsqueda simple de Postulantes	
Tipo de tareas: Desarrollo	Puntos estimados: 1
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde se busca de forma simple o rápida a los postulantes registrados.	

Tarea	
Numero: 6.2	Número historia: 6
Nombre Tarea: Búsqueda avanzada de Postulantes	
Tipo de tareas: Desarrollo	Puntos estimados: 1/2
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde se busca de forma avanzada a los postulantes.	

Tarea	
Numero: 6.3	Número historia: 6
Nombre Tarea: Mostrar detalle del Postulante	
Tipo de tareas: Desarrollo	Puntos estimados: 1/2
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde se muestre el detalle del postulante con toda su información.	

Historia de Usuario	
Numero: 7	Usuario: Postulante
Nombre Historia: Edición de Datos del Postulante	
Prioridad del negocio: Alta	Riesgo en desarrollo: Alta (Alta/Media/Baja)
Puntos estimados: 1	Iteración asignada: 4
Descripción: Se realizará la búsqueda de ofertas Laborales.	
Observaciones: Previo registro de la ofertas laborales.	

Tarea	
Numero: 7.1	Número historia: 7
Nombre Tarea: Edición datos complementarios del postulante	
Tipo de tareas: Desarrollo	Puntos estimados: 1
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde se edite los datos complementarios del postulante.	

Tarea	
Numero: 7.2	Número historia: 7
Nombre Tarea: Edición de los conocimientos o aptitudes del postulante	
Tipo de tareas: Desarrollo	Puntos estimados: 1
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde se edite los conocimientos o aptitudes del postulante.	

Tarea	
Numero: 7.3	Número historia: 7
Nombre Tarea: Edición de la experiencia laboral	
Tipo de tareas: Desarrollo	Puntos estimados: 1
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde se edite la experiencia laboral del postulante.	

Tarea	
Numero: 7.4	Número historia: 7
Nombre Tarea: Edición cursos realizados	
Tipo de tareas: Desarrollo	Puntos estimados: 1
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde se edite los cursos realizados del postulante.	

Módulo Reportes

Historia de Usuario	
Numero: 8	Usuario: Postulante, Institución, Administrador
Nombre Historia: Gráficos estadísticos de conocimientos postulantes	
Prioridad del negocio: Alta	Riesgo en desarrollo: Alta (Alta/Media/Baja)
Puntos estimados: 1	Iteración asignada: 4
Descripción: Se realizará un reporte estadístico sobre los postulantes con conocimientos específicos.	
Observaciones: Previo registro de postulantes con conocimientos.	

Tarea	
Numero: 8.1	Número historia: 8
Nombre Tarea: Reporte estadístico de postulantes con conocimientos	
Tipo de tareas: Desarrollo	Puntos estimados: 1
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde visualice la tendencia de los conocimientos en los postulantes registrados en el sistema.	

Módulo de Administrador

Historia de Usuario	
Numero: 9	Usuario: Administrador
Nombre Historia: Privilegios de usuarios	
Prioridad del negocio: Alta	Riesgo en desarrollo: Alta (Alta/Media/Baja)
Puntos estimados: 1	Iteración asignada: 4
Descripción: Se realizará los permisos respectivos para cada usuario del sistema	
Observaciones: Previo registro de usuarios en el sistema.	

Tarea	
Numero: 9.1	Número historia: 9
Nombre Tarea: Administrar privilegios de usuarios	
Tipo de tareas: Desarrollo	Puntos estimados: 1
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde se administre los privilegios de cada usuario en el sistema.	

Tarea	
Numero: 9.2	Número historia: 9
Nombre Tarea: Cambio de contraseña de usuarios	
Tipo de tareas: Desarrollo	Puntos estimados: 1
Fecha de inicio:	Fecha fin:
Responsable: Richard Pablo Quispe Huiza	
Descripción: Se desarrollara una ventana donde se cambie la contraseña a los usuarios registrados en el sistema.	