
Desarrollo de un Prototipo de Herramienta de Software para la Gestión de Cargas Académicas y
Horarios de la Corporación Universitaria del Caribe—CECAR

Álvaro Enrique Pórtela de la Rosa
Carlos Andrés Tuiran Macías

Corporación Universitaria del Caribe – CECAR
Facultad de Ciencias Básicas, Ingenierías Y Arquitectura
Programa de Ingeniería de Sistemas
Sincelejo - Sucre
2019

Desarrollo de un Prototipo de Herramienta de Software para la Gestión de Cargas Académicas y
Horarios de la Corporación Universitaria del Caribe—CECAR

Álvaro Enrique Pórtela de la Rosa
Carlos Andrés Tuiran Macías

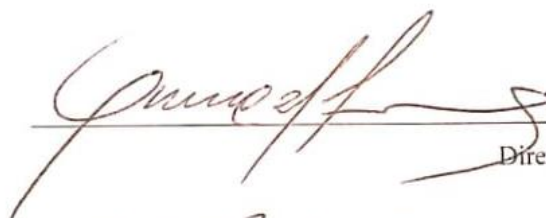
Trabajo de grado presentado como requisito para optar al título de Ingeniero de Sistemas

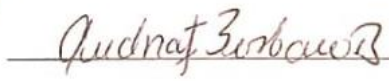
Director
Ing. Guillermo Hernández Hernández

Corporación Universitaria del Caribe – CECAR
Facultad de Ciencias Básicas, Ingenierías Y Arquitectura
Programa de Ingeniería de Sistemas
Sincelejo - Sucre
2019

Nota de Aceptación

4,2


Director


Evaluador 1


Evaluador 2

Sincelejo, Sucre, 3 de mayo 2019

Tabla de contenido

Capítulo I	10
1. Introducción.....	11
Capítulo II.....	13
2. Diseño técnico y metodológico	14
2.1. Fase I: Caracterización de los procesos e identificación de las herramientas y metodología.....	14
2.1.1. Definición de metodología de desarrollo.....	17
2.1.2. Identificación de las tecnologías de la información.....	21
2.2. Fase II: Construcción del prototipo de herramienta de software	22
2.2.1. Historias de Usuario.....	22
2.2.2. Diseño arquitectónico.	26
2.2.3. Diagrama relacional de la base de datos.	27
2.2.4. Planificación.	28
2.2.5. Iteraciones.	28
2.2.5.1. Iteración 1.....	28
2.2.5.2. Iteración 2.....	33
2.2.5.3. Iteración 3.....	37
2.2.5.4. Iteración 4.....	40
2.2.5.5. Iteración 5.....	43
2.2.5.6. Iteración 6.....	46
2.3. Fase III: Evaluación del prototipo de software	51
Capítulo III.....	52
Conclusiones	53
Recomendaciones	54
Referencias bibliográficas.....	55
Anexos	56

Listado de tablas

Tabla 1. Comparación entre los tipos de metodologías de desarrollo de software.....	18
Tabla 2. Aplicaciones y herramientas utilizadas para el desarrollo del Prototipo de Herramienta de Software.	21
Tabla 3. Roles de desarrollo.....	22
Tabla 4. Formato historia de usuario.	23
Tabla 5. Historia de usuario #1.	23
Tabla 6. Historia de usuario #2.	23
Tabla 7. Historia de usuario #3.	24
Tabla 8. Historia de usuario #4.	24
Tabla 9. Historia de usuario #5.	25
Tabla 10. Historia de usuario #6.	25
Tabla 11. Planificación de iteraciones.	28
Tabla 12. Planificación de iteración 1.....	28
Tabla 13. Tarea #1, historia #2.	29
Tabla 14. Tarea #2, historia #2.	29
Tabla 15. Tarea #3, historia #2.	30
Tabla 16. Tarea #4, historia #2.	30
Tabla 17. Pruebas de historia de usuario #2.	33
Tabla 18. Planificación de iteración 2.....	34
Tabla 19. Tarea #5, historia #1.	34
Tabla 20. Tarea #6, historia #1.	34
Tabla 21. Tarea #7, historia #1.	35
Tabla 22. Pruebas de historia de usuario #1.	37
Tabla 23. Planificación de iteración 3.....	37
Tabla 24. Tarea #8, historia #3.	37
Tabla 25. Tarea #9, historia #3.	38
Tabla 26. Tarea #10, historia #3.	38
Tabla 27. Pruebas de historia de usuario #3.	39
Tabla 28. Planificación de iteración 4.....	40
Tabla 29. Tarea #11, historia #4.	40
Tabla 30. Tarea #12, historia #4.	41
Tabla 31. Tarea #13, historia #4.	41
Tabla 32. Pruebas de historia de usuario #4.	43
Tabla 33. Planificación de iteración 5.....	43
Tabla 34. Tarea #14, historia #6.	44
Tabla 35. Tarea #15, historia #6.	44
Tabla 36. Pruebas de historia de usuario #6.	46
Tabla 37. Planificación de iteración 6.....	46
Tabla 38. Tarea #16, historia #5.	46

Tabla 39. Tarea #17, historia #5.	47
Tabla 40. Tarea #18, historia #5.	47
Tabla 41. Pruebas de historia de usuario #5.	50

Listado de figuras

<i>Figura 1.</i> Flujograma coordinación de programa	16
<i>Figura 2.</i> Flujograma Vicerrectoría académica	17
<i>Figura 3.</i> Diseño arquitectónico	26
<i>Figura 4.</i> Diagrama relacional de la base de datos	27
<i>Figura 5.</i> Vista para creación de programa académico.	31
<i>Figura 6.</i> Vista donde se listan los diversos programas académicos.....	32
<i>Figura 7.</i> Vista donde se listan los diversos pensum de un programa académico.....	32
<i>Figura 8.</i> Vista donde se listan las asignaturas de un pensum.....	33
<i>Figura 9.</i> Vista de creación de docentes.	36
<i>Figura 10.</i> Vista donde se listan los docentes.....	36
<i>Figura 11.</i> Vista de edición de docentes.....	36
<i>Figura 12.</i> Vista de listado de periodos.	39
<i>Figura 13.</i> Vista de la creación/edición de periodos.	39
<i>Figura 14.</i> Vista de la creación de grupos.	42
<i>Figura 15.</i> Vista donde se listan los grupos.....	42
<i>Figura 16.</i> Vista para la edición los grupos.....	43
<i>Figura 17.</i> Vista para la creación de novedades.	45
<i>Figura 18.</i> Vista donde se listan las novedades.....	45
<i>Figura 19.</i> Componente de asignación de horarios.	48
<i>Figura 20.</i> Comparación de carga actual con aprobada.	49
<i>Figura 21.</i> Vista para exportar horario.	49
<i>Figura 22.</i> Ejemplo de horario exportado.....	50

Resumen

El presente trabajo desarrolla un prototipo funcional para la gestión de cargas académicas utilizando herramientas con licencia de software libre para mejorar los procesos al interior de las diferentes dependencias involucradas en dicho proceso en la Corporación Universitaria del Caribe –CECAR.

Este proceso de desarrollo de software toma en cuenta los problemas presentados al momento de construir los horarios y cargas académicas tanto para docentes como para estudiantes, la herramienta es un prototipo funcional a modo de aplicación web que apoya los métodos y artefactos de software pre existentes dentro de la corporación guiando y administrando las etapas del proceso en cuestión permitiendo a su vez agilizar la toma de decisiones para el personal involucrado.

Para la implementación del prototipo se utilizó la metodología de la programación extrema o XP, la cual facilita la construcción de la herramienta y permite lograr la culminación satisfactoria de las necesidades propias de los usuarios. Se especificaron las fases de planificación, análisis, diseño, codificación y pruebas.

Se hizo un seguimiento al proceso, al igual que se analizó su implicación para adaptarse en la corporación universitaria, el trabajo de desarrollo durante un ciclo académico en donde se observó la generación de las cargas académicas de algunos programas.

Palabras clave: horarios, generación de novedades, carga académica, procesos desarrollo de software, proyección de grupos.

Abstract

The present work develops a functional prototype for the management of academic loads using tools with free software license to improve the processes within the different dependencies involved in this process in the Caribbean University Corporation -CECAR.

This software development process takes into account the problems presented when constructing schedules and academic loads for both teachers and students, the tool is a functional prototype as a web application that supports pre-existing software methods and artifacts within of the corporation guiding and administering the stages of the process in question, allowing at the same time to expedite decision-making for the personnel involved.

For the implementation of the prototype the methodology of extreme programming or XP was used, which facilitates the construction of the tool and allows to achieve the satisfactory completion of the user's own needs. The phases of planning, analysis, design, coding and testing were specified.

The process was followed up, as was its implication to adapt in the university corporation, the development work during an academic cycle where the generation of the academic loads of some programs was observed.

Keywords: schedules, generation of novelties, academic load, processes

Software development, group projection.

Capítulo I

1. Introducción

El impacto de las tecnologías, principalmente las de la Información es notorio en las organizaciones. Las Tecnologías agregan valor a las labores operativas y de gestión administrativa en general y permite a las organizaciones mejorar y optimizar sus procesos internos con el fin de centrarse en actividades propias de su misión como organización. El objetivo de las tecnologías de la información en una organización consiste en procesar, almacenar, difundir y gestionar todo tipo de información en las distintas unidades o departamentos de una organización.

El presente proyecto de desarrollo tecnológico se refiere a un prototipo de herramienta de software de gestión de cargas académicas y de horarios de docentes y estudiantes destinado al apoyo parcial del proceso de planeación curricular que se lleva a cabo cada semestre y que pretende más específicamente apoyar las labores de gestión de las cargas académicas que se desarrollan al interior de la Corporación Universitaria del Caribe – CECAR. Esta herramienta realizará operaciones relacionadas con la proyección de grupos, asignación de carga académica, asignación de horarios, generación de novedades, entre otras funcionalidades. Este prototipo es viable económicamente, dado que se utilizará herramientas con licencia de software libre para su desarrollo, además de que si se lleva a cabo su implantación se utilizaría la infraestructura tecnológica de la corporación, lo cual conllevaría a que los costos se reduzcan considerablemente.

Para analizar esta problemática es necesario mencionar sus causas. Meses antes al inicio de cada semestre académico, los decanos y coordinadores académicos de cada uno de los programas ofrecidos por CECAR, inician el proceso de proyección de grupos y asignación de horarios a profesores y estudiantes, estos se ven enfrentados a diversos inconvenientes que retrasan o hacen más complicado dicho proceso, inconvenientes como el número mínimo de estudiantes por curso ya que si no se cuenta con los suficientes un curso no puede ser abierto, los cruces entre horarios, asignaturas y docentes, sumado a las dificultades de comunicar dichos cruces con formatos en hojas de cálculo limitados a una sola estación de trabajo

No solo las directivas se van afectadas a la hora de la matrícula, también los estudiantes debido a los retrasos en la generación de una nueva configuración de horarios. La creación de

grupos nuevos implica la asignación de un profesor, lo que conlleva tal vez una reacomodación total o parcial de los horarios ya establecidos.

Entonces se podría decir que se reinicia de nuevo el trabajo de las coordinaciones académicas encargadas de la asignación de la carga académica.

Actualmente el proceso de planificación de cargas académicas se realiza manualmente principalmente a través de formatos en hojas de Excel y apoyados por el sistema de autenticación único ALDEA. Estos documentos son cargados a los módulos ofrecidos por ALDEA, la cual es donde reposara la carga al final del proceso

Estos procedimientos generan inconvenientes que hacen que dicho proceso sea lento y tedioso de realizar, además de producir un alto número de inconsistencias en la información las cuales a su vez no son fáciles de atender y corregir, como las solicitudes de docentes por parte de otros programas o los cruces de docentes entre facultades.

Se proyectó que el prototipo contribuyera a brindar un acercamiento más simple, rápido y de primera mano con miras a la realización de una planificación más organizada por parte de las diversas Facultades y Coordinaciones Académica de los diversos programas que ofrece la Corporación previa al reporte final cargado a la plataforma ALDEA.

Cabe resaltar que para la realización del proyecto se contó con el apoyo de la facultad de ciencias básicas, ingenierías y arquitectura la cual a través de sus coordinadores serían usuarios principales de la herramienta.

Capítulo II

2. Diseño técnico y metodológico

Para el correcto seguimiento y ejecución de la construcción de una herramienta software y en general para cualquier proyecto se hace necesario un norte o una guía y es aquí donde entran a mediar las metodologías de desarrollo de software. Con respecto a metodología “La palabra **metodología** tiene su origen en el idioma griego, y se refiere al modelo aplicable que deben necesariamente seguir los métodos de investigación, aun cuando resulten cuestionables”. (Enciclopedia de Conceptos, 2018. "Metodología". Recuperado de: <https://concepto.de/metodologia/>, párr.1)

2.1. Fase 1: Caracterización de los procesos e identificación de las herramientas y metodología

Para lograr entender y explorar las necesidades del cliente se utilizó como mecanismo de recolección de información una entrevista de tipo abierta con uno de los coordinadores de la corporación realizada y grabada en el año 2016. “Una entrevista abierta es una técnica cualitativa de investigación en la cual se propicia una conversación no estructurada entre una persona que entrevista y un entrevistado. Está basada en preguntas generadas espontáneamente como parte de la interacción comunicacional” (Corvo T, 2018. Entrevista Abierta: Tipos, Ventajas, Desventajas y Ejemplos. Recuperado de <https://www.lifeder.com/entrevista-abierta/>, párr. 1)

Antes de realizar la entrevista se tenía de antemano en mente conocer el proceso de generación de cargas académicas en su totalidad y más concretamente la pregunta principal que era ¿Cómo se lleva a cabo el proceso de cargas académicas dentro de la corporación? Pero en el transcurso de la entrevista surgieron las siguientes preguntas:

- ✓ ¿Qué es una carga académica?
- ✓ ¿Cuánto tiempo toma realizarla?
- ✓ ¿Cuál es el motivo de realizar la carga académica?
- ✓ ¿Cómo es el procedimiento para realizar una carga académica?

- ✓ ¿Qué información debe recibir en particular de otras personas y/o departamentos para realizar la carga académica?
- ✓ ¿Cómo debe presentarse al final el resultado del proceso de carga académica?
- ✓ ¿Qué persona o dependencia recibe los datos una vez finalizado su trabajo?
- ✓ ¿Poseen algún programa informático con el cual realizan esta tarea?
- ✓ ¿Qué cosas considera que son más complicadas actualmente y que se puede mejorar?
- ✓ ¿Hay alguna necesidad especial que requiera implementarse?
- ✓ ¿El resultado final está sujeto a cambios?

La entrevista arrojó que el proceso de cargas académicas se lleva a cabo con el sistema de gestión académica ofrecido a través de ALDEA apoyado de manera manual a través de formatos de Excel, uno donde reposa la carga y otro para el reporte de novedades (Ver Anexos A y B) como también se describe el proceso de creación de una carga académica el cual inicia cuando se verifica la planta docente y la cantidad de alumnos matriculados por semestre, acto seguido se procede a hacer la planeación grupal por semestre y asignaturas, una vez culmina este proceso se realiza la asignación de cada grupo por asignatura a un docente idóneo para impartirla e inmediatamente se prosigue con la asignación de un horario dentro de los rangos establecidos por la corporación.

Teniendo ya una carga preliminar definida se publica para recibir las novedades pertinentes por parte de docentes y luego de dar a lugar a estas pasa a vicerrectoría académica para someterse al proceso de aprobación o corrección según corresponda, cabe resaltar que una vez se de aprobación a una carga esta debe reposar inmodificable y en caso de necesitar modificaciones (novedades) estas deben aplicarse a una copia de la carga ya aprobada, esto con el fin de llevar un histórico de las modificaciones a las que ha sido sometida una carga original. El proceso finaliza con la publicación de carga final a cada docente y estudiantes discriminada por semestre (Ver figuras 1 y 2)

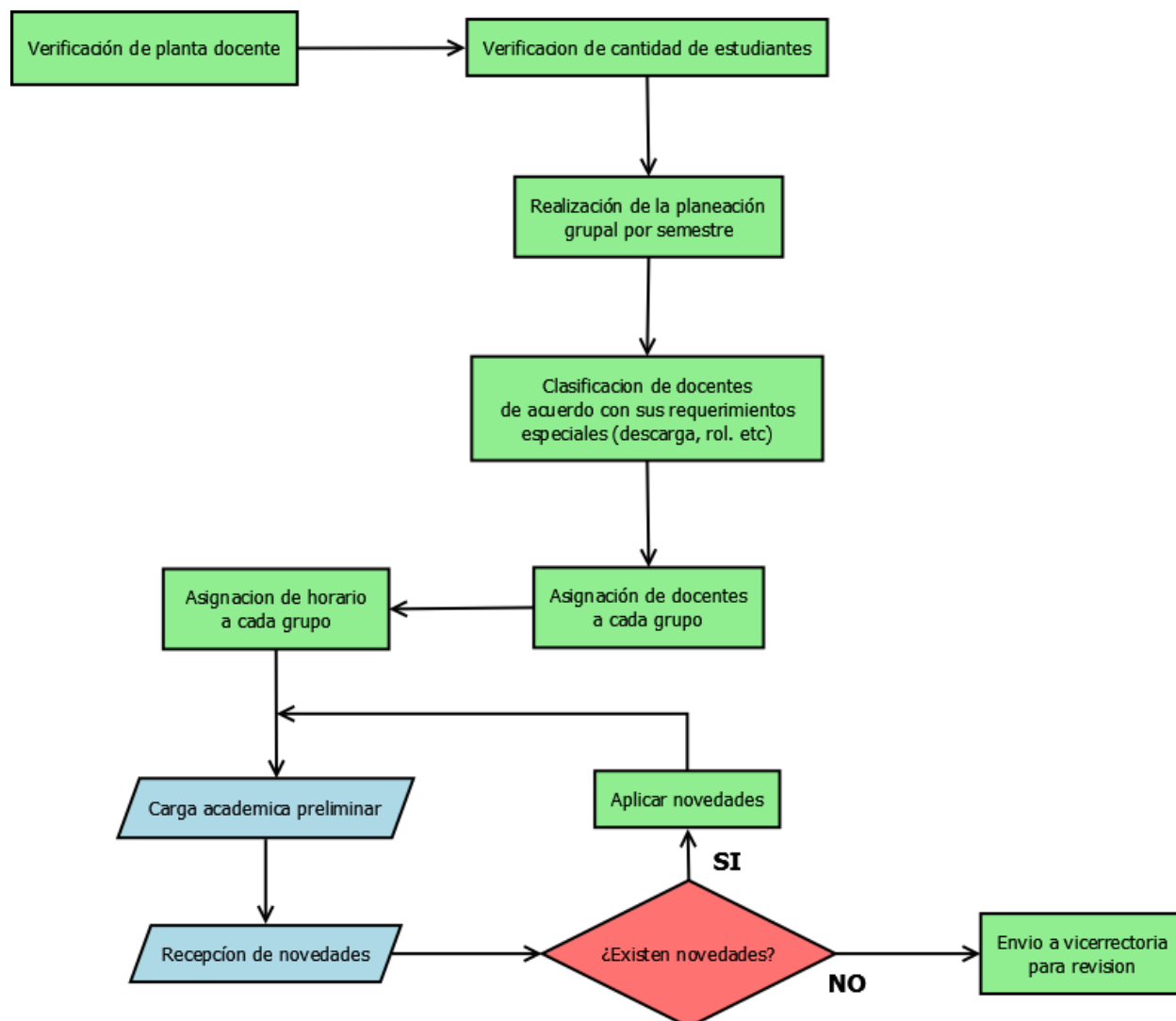


Figura 1. Flujograma coordinación de programa

Fuente: Elaboración propia

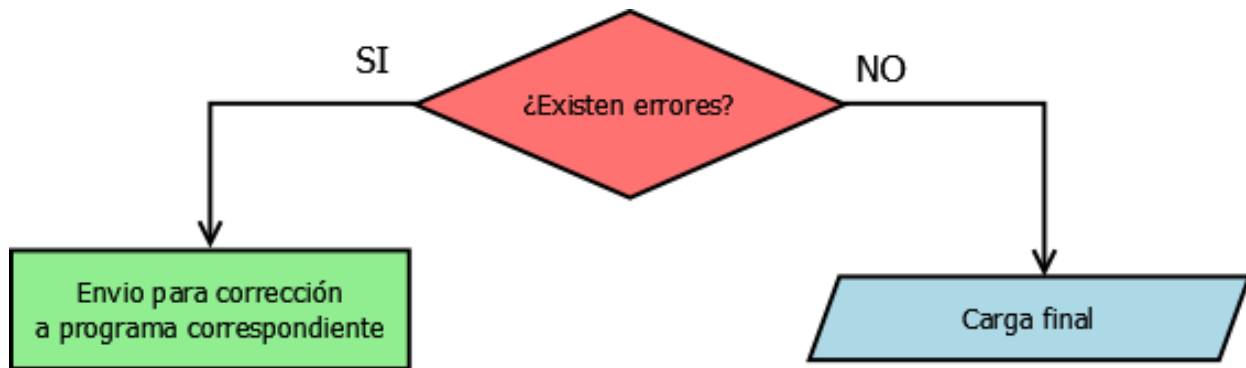


Figura 2. Flujograma Vicerrectoría académica
Fuente: Elaboración propia

2.1.1. Definición de metodología de desarrollo.

En todo proceso de desarrollo de software se hace necesario pasar a través de varias fases que nos llevan a la culminación exitosa de un producto de software totalmente funcional y útil.

En el contexto de la ingeniería de software, un proceso no es una prescripción rígida de cómo elaborar software de cómputo. Por el contrario, es un enfoque adaptable que permite que las personas que hacen el trabajo (el equipo de software) busquen y elijan el conjunto apropiado de acciones y tareas para el trabajo. Se busca siempre entregar el software en forma oportuna y con calidad suficiente para satisfacer a quienes patrocinaron su creación y a aquellos que lo usarán. (Pressman, 2010, p. 12)

Ahora bien, si dentro de la ingeniería del software la construcción de artefactos de software no es un proceso rígido como mencionábamos anteriormente esto no significa que no pueda existir una estructura generalizada de actividades dentro del proceso que sirva como punto de partida para las diversas metodologías de desarrollo que han surgido a lo largo de la historia, dicha estructura cuenta con 5 actividades puntuales a saber:

- ✓ **Comunicación:** se debe realizar esta actividad tanto con clientes como con todos los actores involucrados dentro de la construcción de la herramienta software de manera

efectiva en miras de entender cómo debería funcionar dicho software y que características debería poseer.

- ✓ **Planeación:** dado que en todo proyecto se necesita un norte que seguir es en esta actividad donde se construye una especie de mapa que define riesgos, tareas a realizar, como realizarlas y en qué momento dentro del contexto del software y así lograr su exacta implementación.
- ✓ **Modelado:** si en la actividad anterior construíamos el mapa de nuestra herramienta software, en el modelado lo que se busca es realizar un esquema donde se muestre como funcionaría el software y como se relacionarían sus diversos componentes.
- ✓ **Construcción:** comprende el hecho de producción del código y las pruebas respectivas del software.
- ✓ **Despliegue:** dicese de realizar al cliente la entrega de una mejora o del producto total terminado con el fin de usarse o de recibir retroalimentación para mejoras posteriores.

En el campo del desarrollo del software podemos hablar de dos tipos de metodologías, las metodologías tradicionales y la metodología agiles, la primera se enfoca principalmente en procesos asociados a la planeación y control del proyecto con un alto detalle en la especificación de requisitos del sistema claramente definidos, imponiendo la metodología sobre el proceso de desarrollo ya que se caracterizan por un riguroso trabajo a nivel de roles y requisitos.

Por otra parte, las segundas se enfocan en procesos incrementales con entregas funcionales ya que buscan aumentar la confianza de los clientes al vincularlos en el proceso mediante cooperación entre estos y el equipo de trabajo, se caracterizan por reducir la cantidad de documentación y están basadas en el manifiesto ágil.

Como podemos ver a continuación (Ver tabla 1) las diferencias entre los dos grandes grupos de metodologías de desarrollo de software son bastante significativas y determinan el rumbo de un proyecto de desarrollo de software.

Tabla 1.

Comparación entre los tipos de metodologías de desarrollo de software

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos Más artefactos	Pocos roles Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

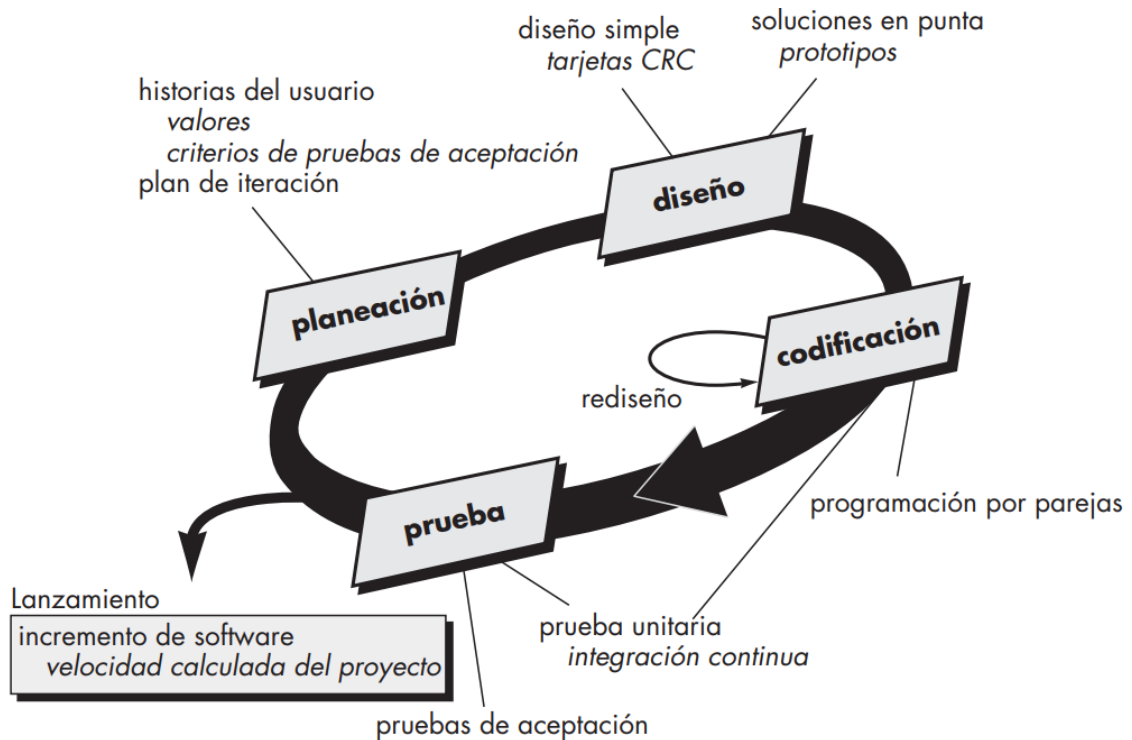
Fuente: Rivadeneira. (2012). metodologías ágiles enfocadas al modelado de requerimientos.

Teniendo en cuenta todo lo anterior y analizando las condiciones de nuestro proyecto, este es de alcance medio, por lo que el nivel de especificidad no es muy elevado, adicional a esto el equipo de desarrollo es reducido y debido a la naturaleza compleja de la lógica del negocio detrás del proyecto se requiere que se hagan liberaciones periódicas para que el usuario final pueda apreciar los cambios gradualmente y así someter a correcciones inmediatas mediante un proceso iterativo.

Por lo tanto, se reúnen todas las condiciones para decantarnos por una metodología ágil que cubra las especificaciones antes mencionadas, puesto que una metodología tradicional estaría fuera de contexto porque el tamaño de los equipos, el nivel de especificidad y alcance está pensado para proyectos más grandes. La metodología ágil que más encaja a estas especificaciones sería programación extrema por su fácil implementación y construcción iterativa según se enuncia en su proceso y características.

La programación extrema usa un enfoque orientado a objetos como paradigma preferido de desarrollo, y engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y

pruebas. La figura 3.2 ilustra el proceso XP y resalta algunas de las ideas y tareas clave que se asocian con cada actividad estructural. (Pressman 2010, p. 62)



Siguiendo lo que se enuncia en el gráfico del proceso de la metodología XP, esta toma las 5 actividades del modelo general antes mencionado y las redefine siguiendo el manifiesto ágil para poder denominarse una metodología ágil en pleno como se describe a continuación.

Planificación: dentro la metodología XP la planificación se compone de la tarea explorar y entender en conjunto con el cliente el contexto del proyecto para que con esta información obtenida los miembros del equipo de desarrollo puedan construir las llamadas “historias de usuario”, lo que se comportaría como los requerimientos del sistema en esta metodología y que definen funcionalidades puntuales a codificar, posteriormente las historias de usuario se agrupan por prioridad para componer lo que se conoce como iteración. Al final de cada iteración se tendrá una versión parcial de la herramienta software cuya funcionalidad ira incrementando según se vayan implementando historias de usuario en las diferentes iteraciones.

Diseño: como nos dice Pressman (2010) “El diseño XP sigue rigurosamente el principio MS (mantenlo sencillo). Un diseño sencillo siempre se prefiere sobre una representación más compleja. Además, el diseño guía la implementación de una historia conforme se escribe: nada más y nada menos” (p. 63). Dicha guía de implementación de una historia se logra usando una herramienta llamada tarjetas CRC (clase-responsabilidad colaborador) que usan para llevar las historias de usuario a un formato de clase que representa un objeto computacional.

Codificación: para la codificación XP se usa principalmente la programación en parejas, que no es más que dos personas trabajen en una misma estación para la construcción de una única historia de usuario con el fin de evitar la mayor cantidad de errores y tener una visión más holística de esta ya que mientras una persona codifica la otra puede ir observando los detalles que se le escapen a la persona que codifica o verificar que se esté codificando de manera correcta la historia en cuestión.

Pruebas: el proceso de pruebas en la metodología XP suele llevarse a través de las pruebas unitarias y se busca automatizarlas a través de una herramienta ya existente. En conjunto con estas también se usan las pruebas de integración y validación de manera constante para indicar de manera temprana fallas que necesiten ser corregidas. Ya por último se utilizan pruebas de aceptación que son definidas por el cliente, quien establece como debe comportarse cada una de las funcionalidades que el como cliente puede realizar y por tanto revisar.

2.1.2. Identificación de las tecnologías de la información.

Tabla 2.

Aplicaciones y herramientas utilizadas para el desarrollo del Prototipo de Herramienta de Software.

Herramienta de desarrollo	Utilidad	Versión
Visual Studio	Entorno de desarrollo	Community 2017
SQL Server	Motor de bases de datos	2016
Windows	Sistema Operativo	10
SQL Server Management Studio	Administrador Gráfico de Bases de Datos	2017

ASP.NET	Framework de desarrollo web	5.0
C#	Lenguaje de Programación	
Microsoft Word	Editor de texto	2016

Fuente: Elaboración propia

2.1.4. Establecimiento de roles.

Tabla 3.

Roles de desarrollo

Responsable	Rol	Contacto
Álvaro Enrique Pórtela de la Rosa	Desarrollador	alvaro.portelad@cecar.edu.co
Carlos Andrés Tuiran Macías	Desarrollador, tester	carlos.tuiranm@cecar.edu.co

Fuente: Elaboración Propia

2.2. Fase II: Construcción del prototipo de herramienta de software

2.2.1. Historias de Usuario.

Dentro del proceso de la metodología XP se encuentra una fase exploratoria que es de vital importancia para el levantamiento de las necesidades que suplirá el prototipo de software a realizar la forma de obtener dicha información como se había mencionado en secciones anteriores fue la entrevista abierta de la cual se extrajeron datos importantes de las preguntas que surgieron estos datos deben ser transformados en requerimientos más explícitos para su posterior codificación.

Dentro de la metodología XP encontramos una poderosa y simple herramienta para convertir los datos recolectados en información, dicha herramienta son las historias de usuarios que no son más que formatos que se alimentan de datos precisos y relevantes de manera rápida, coherente y concisa por parte de un usuario final.

Una historia de usuario es aquella que puede escribirse con la siguiente frase:

Como [un usuario], **puedo** [acción/funcionalidad] **para** [beneficio]

Por ejemplo: Como administrador del sistema, puedo agregar productos al catálogo para ser visualizados por los clientes. Muchas veces, puede resultar redundante o hasta incluso carecer de sentido, indicar el beneficio. Por ello, es frecuente describir las historias de usuario, sin incorporar este tercer elemento: Como administrador del sistema, puedo agregar productos al catálogo. (Bahit 2012, p. 42).

Tabla 4.

Formato historia de usuario.

<i>Historias de usuario</i>			
No.		Usuario:	
Nombre de Historia:			
Prioridad en negocio:	Riesgo:	Dificultad:	Iteración:
Descripción:			
Observaciones:			

Tabla 5.

Historia de usuario #1.

<i>Historias de usuario</i>			
No. 1		Usuario: Coordinador de Programa	
Nombre de Historia: Gestionar Docentes			
Prioridad en negocio: 5	Riesgo: Bajo	Dificultad: Baja	Iteración: 2
Descripción: <ul style="list-style-type: none"> • Como coordinador debo poder gestionar los docentes • Los datos de los docentes para efectos de la generación del horario solo serán: el nombre completo, su documento de identidad y el tipo de contratación. 			
Observaciones: <ul style="list-style-type: none"> • la eliminación y la carga masiva de docentes se hará vía base de datos sujeta a aprobación por parte del coordinador 			

Tabla 6.

Historia de usuario #2.

<i>Historias de usuario</i>			
No. 2		Usuario: Coordinador de programa	
Nombre de Historia: Gestionar programas			
Prioridad en negocio: 6	Riesgo: Alto	Dificultad: Alta	Iteración: 1
Descripción:			
<ul style="list-style-type: none"> • Quiero poder gestionar las asignaturas que se impartirán en mi programa académico • Debo poder gestionar pensum diferentes • De cada asignatura se debe tener un código, nombre, semestre e intensidad horaria • Dichas asignaturas deben asignarse a un pensum previamente creado y poder seleccionar un pensum según el caso 			
Observaciones:			
<ul style="list-style-type: none"> • la eliminación y la carga masiva de asignaturas y/o pensum se hará vía base de datos sujeta a aprobación por parte del coordinador 			

Tabla 7.

Historia de usuario #3.

<i>Historias de usuario</i>			
No. 3		Usuario: Coordinador de Programa	
Nombre de Historia: Gestionar periodos académicos (semestres)			
Prioridad en negocio: 4	Riesgo: Alto	Dificultad: Media	Iteración: 3
Descripción:			
<ul style="list-style-type: none"> • Se deben poder crear periodos académicos para agrupar la respectiva carga en ellos • Deben ser creados con un identificador y nombre • Se debe poder editar el nombre de los periodos 			
Observaciones:			
<ul style="list-style-type: none"> • La eliminación de periodos solo se hará vía base de datos sujeta a aprobación por parte del coordinador 			

Tabla 8.

Historia de usuario #4.

<i>Historias de usuario</i>	
No. 4	Usuario: Coordinador de Programa

Nombre de Historia: Gestionar Grupos			
Prioridad en negocio: 3	Riesgo: Medio	Dificultad: Alta	Iteración: 4
Descripción:			
<ul style="list-style-type: none"> • El prototipo debe permitir gestionar grupos de estudiantes por cada asignatura del pensum seleccionado. • Deben tener un número, número de estudiantes, programa, semestre, asignatura y un docente tomado de los ya registrados previamente para el programa 			
Observaciones:			
<ul style="list-style-type: none"> • La eliminación de grupos solo se hará vía base de datos sujeta a aprobación por parte del coordinador 			

Tabla 9.

Historia de usuario #5.

<i>Historias de usuario</i>			
No. 5		Usuario: Coordinador/Docentes	
Nombre de Historia: Gestión de Horarios			
Prioridad en negocio: 1	Riesgo: Alto	Dificultad: Alta	Iteración: 6
Descripción:			
<ul style="list-style-type: none"> • Se debe poder asignar un horario a cada grupo de estudiantes creados previamente para cada asignatura teniendo en cuenta su intensidad horaria y la disponibilidad tanto de estudiantes como de docentes • El prototipo debe permitir a los coordinadores exportar los horarios para los docentes y estudiantes en formato Excel • Se debe poder ir guardando una versión parcial de la carga académica con un botón de aprobación de carga 			
Observaciones:			

Tabla 10.

Historia de usuario #6.

<i>Historias de usuario</i>			
No. 6		Usuario: Coordinador de Programa	
Nombre de Historia: Gestión de Novedades			
Prioridad en negocio: 2	Riesgo: Bajo	Dificultad: Baja	Iteración: 5

Descripción:

- Se debe permitir reportar y listar las diversas novedades que se presenten en la carga académica tomando en cuenta como modelo el formato de Excel ya existente para su construcción en el prototipo

Observaciones:

2.2.2. Diseño arquitectónico.

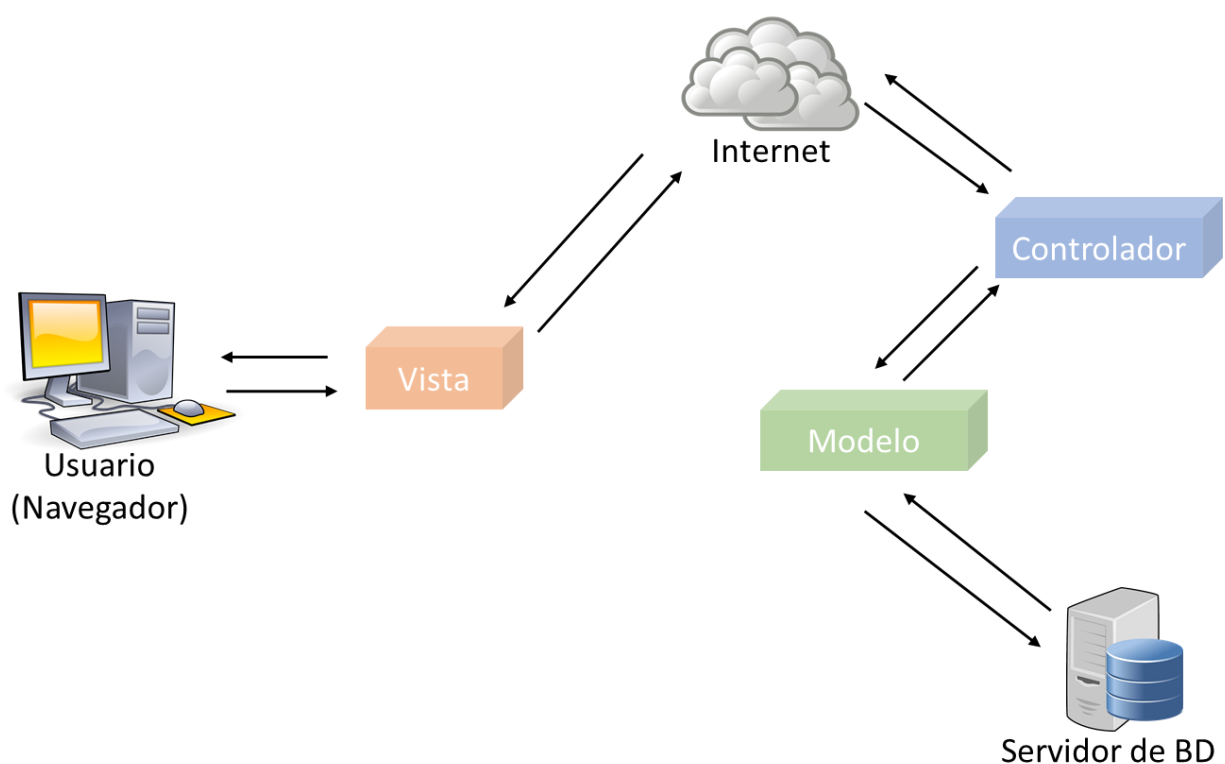


Figura 3. Diseño arquitectónico

Fuente: Elaboración propia

2.2.3. Diagrama relacional de la base de datos.

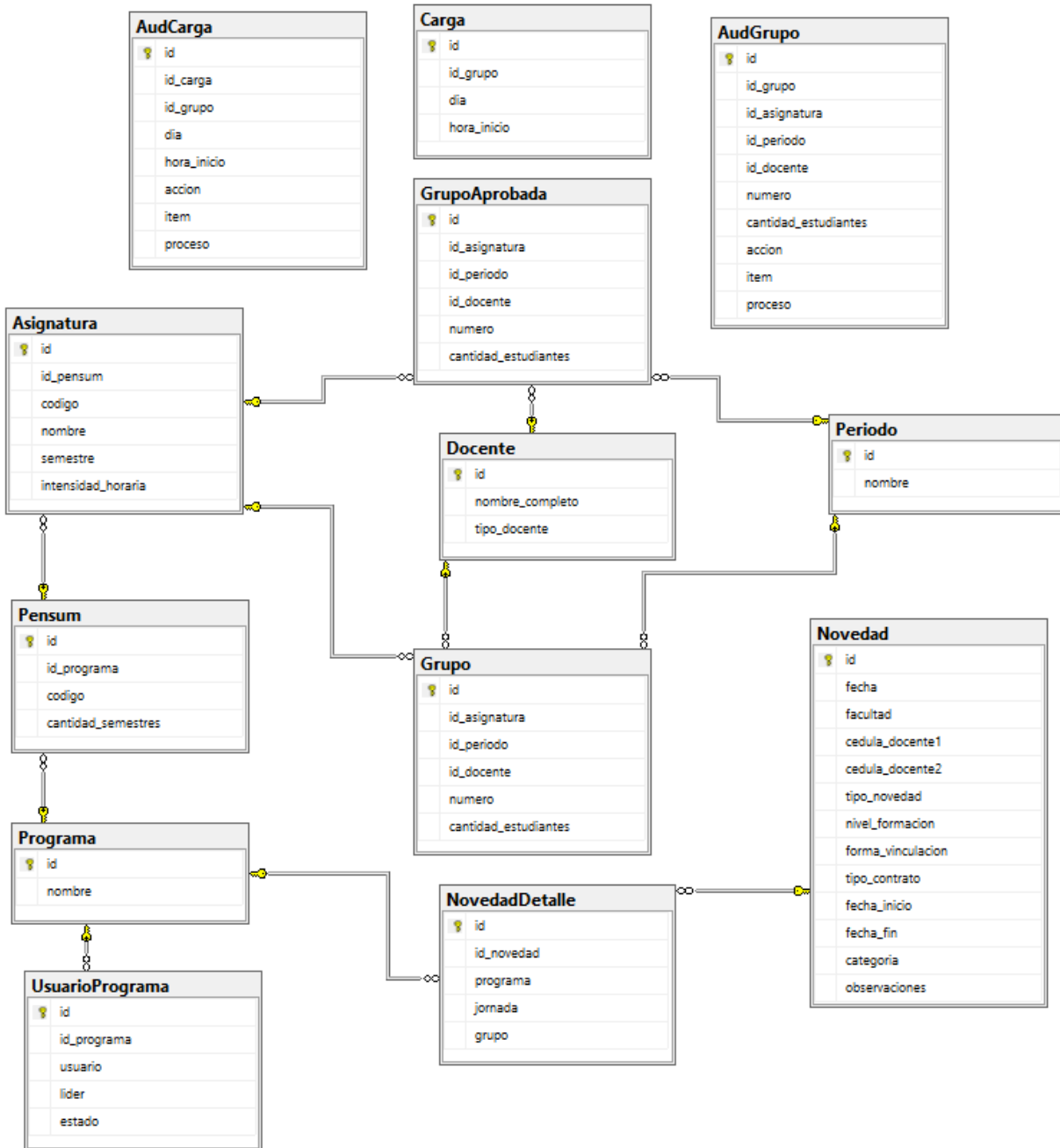


Figura 4. Diagrama relacional de la base de datos

Fuente: Elaboración propia

2.2.4. Planificación.

De acuerdo a la necesidad como prerrequisito para una iteración superior y teniendo en cuenta la complejidad de las historias de usuario se planifico su desarrollo de manera iterativa (Ver tabla 11), dividiendo dichas iteraciones en tareas más específicas siguiendo el proceso de desarrollo de la metodología XP, estas tareas serán descritas con más detalle en secciones posteriores. Dichas iteraciones deben ir arrojando una pieza de software con funcionalidad parcial lo cual permite una mejor retroalimentación y una mayor satisfacción por parte de los usuarios.

Tabla 11.

Planificación de iteraciones.

N°	Nombre de Historia	Prioridad	Riesgo	Dificultad	Iteración
1	Gestionar Docentes	5	Bajo	Baja	2
2	Gestionar Programas	6	Alto	Media	1
3	Gestionar Periodos Académicos	4	Alto	Media	3
4	Gestionar Grupos	3	Medio	Alta	4
5	Gestión de Horarios	1	Alto	Alta	6
6	Gestión de Novedades	2	Bajo	Baja	5

2.2.5. Iteraciones.

A continuación, se presenta totalmente discriminadas las iteraciones en sus respectivas tareas y con sus respectivas pruebas como se había mencionado anteriormente.

2.2.5.1. Iteración 1.

Tabla 12.

Planificación de iteración 1.

N° de historia de usuario	N° de tarea	Nombre de la tarea
2	1	Crear programas académicos.

2	2	Listar programas académicos registrados.
2	3	Gestionar pensum
2	4	Gestionar asignaturas
Lenguajes Usados		SQL, C#, HTML, JavaScript.
Herramientas Usadas		Visual Studio, SQL Management Studio Express.
Tiempo estimado de desarrollo		3 semana.

Tabla 13.

Tarea #1, historia #2.

Formato de tarea	
N° de tarea: 1	N° de historia: 2
Nombre de la tarea: Crear programas académicos.	
Tipo: Desarrollo	Puntos estimados: 3
Responsable: Equipo de desarrollo XP	
Descripción:	
<ul style="list-style-type: none"> • Se debe hacer una vista con el campo nombre y un botón de registro para la creación de los programas académicos. • Se debe dar funcionalidad a dicho botón de registro para almacenar en la base de datos el programa a crear. 	

Tabla 14.

Tarea #2, historia #2.

Formato de tarea	
N° de tarea: 2	N° de historia: 2
Nombre de la tarea: Listar programas académicos registrados.	

Tipo: Desarrollo	Puntos estimados: 4
Responsable: Equipo de desarrollo XP	
Descripción:	
<ul style="list-style-type: none"> • Crear una función que liste los programas registrados desde un apartado del menú en una vista que contenga una tabla donde cada fila represente un programa académico. 	

Tabla 15.

Tarea #3, historia #2.

Formato de tarea	
N° de tarea: 3	N° de historia: 2
Nombre de la tarea: Gestionar pensum	
Tipo: Desarrollo	Puntos estimados: 4
Responsable: Equipo de desarrollo XP	
Descripción:	
<ul style="list-style-type: none"> • Crear una vista donde se listen los pensum de cada programa académico y tenga un botón de crear y uno de editar para cada pensum. • El botón crear debe llevar a una vista donde se creen los pensum con su código y cantidad de semestres y a su vez el botón editar debe permitir cambiar estos campos en una vista similar a la de crear. • Enlazar la vista principal de los pensum al listado de programas a través de un botón para cada uno. 	

Tabla 16.

Tarea #4, historia #2.

Formato de tarea	
N° de tarea: 4	N° de historia: 2
Nombre de la tarea: Gestionar asignaturas	
Tipo: Desarrollo	Puntos estimados: 4

Responsable: Equipo de desarrollo XP

Descripción:

- Crear una vista donde se listen las asignaturas de un pensum de un programa académico y que tenga un botón para crear y otro para editar las asignaturas.
- Los botones de crear y editar deben llevar a una vista donde se permita crear o editar los campos debe tener los campos código, nombre, semestre e intensidad horaria respectivamente.
- Enlazar la vista principal de las asignaturas a la de listado de pensum por medio de un botón para cada pensum.

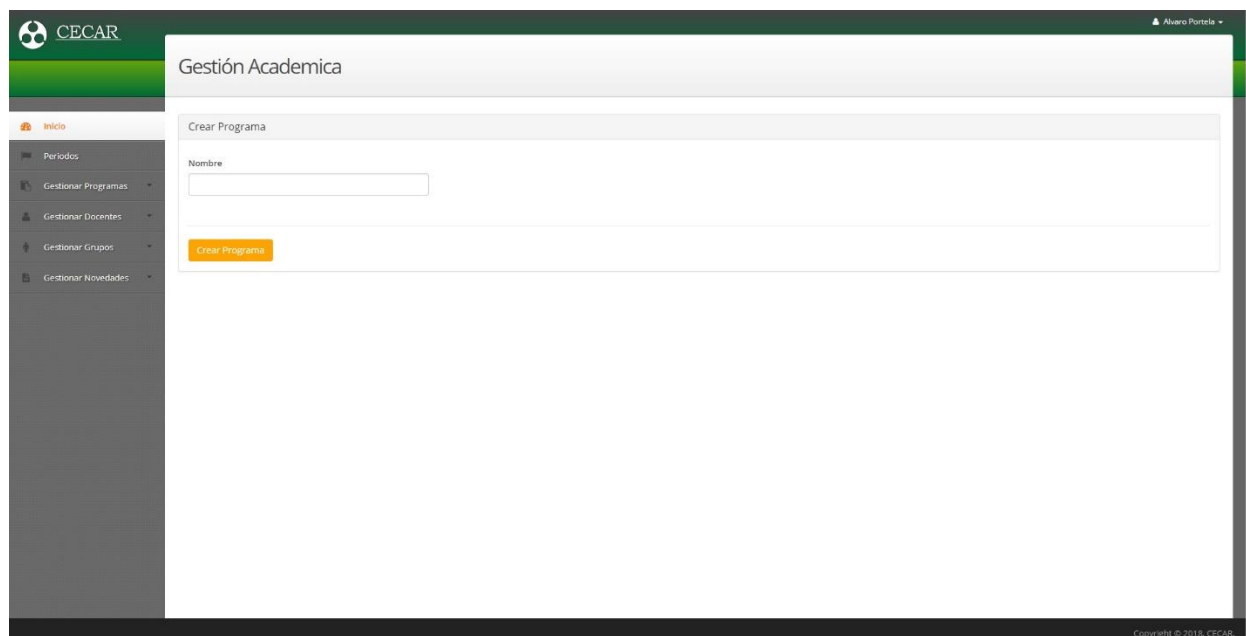


Figura 5. Vista para creación de programa académico.

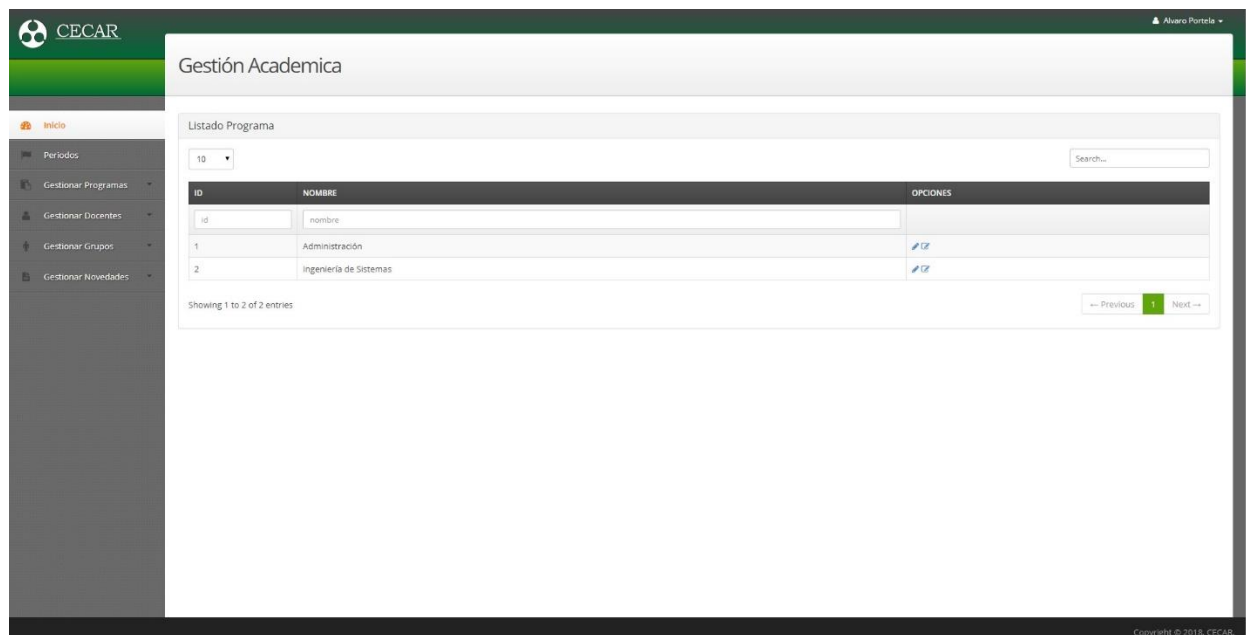


Figura 6. Vista donde se listan los diversos programas académicos.

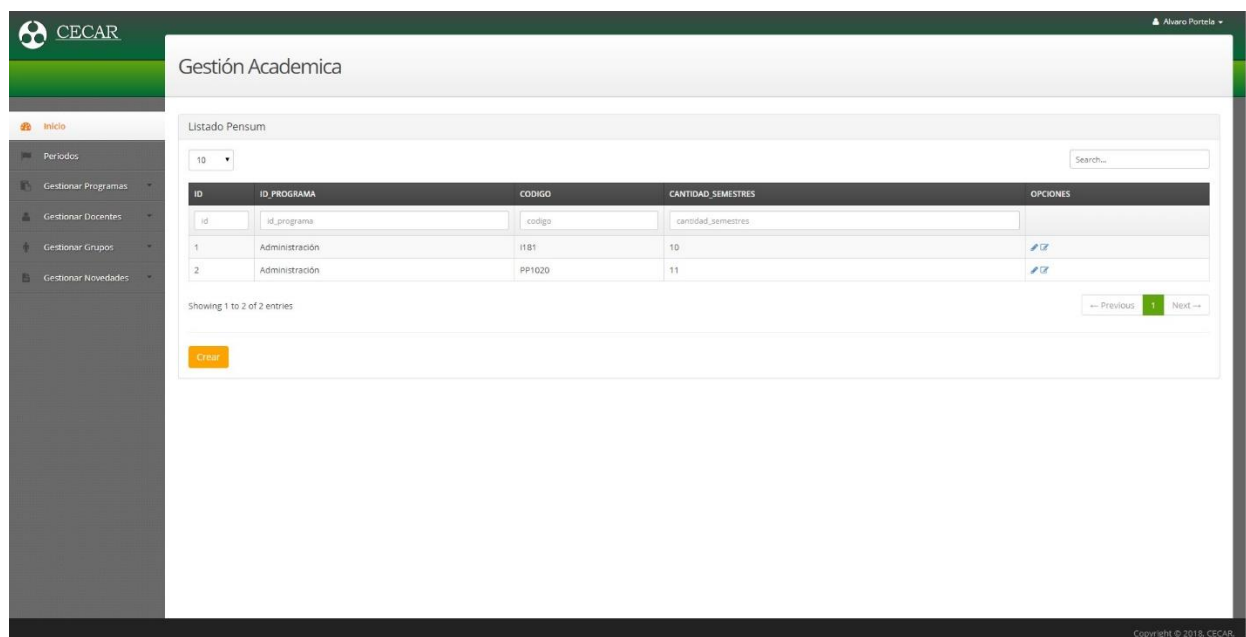


Figura 7. Vista donde se listan los diversos pensum de un programa académico.

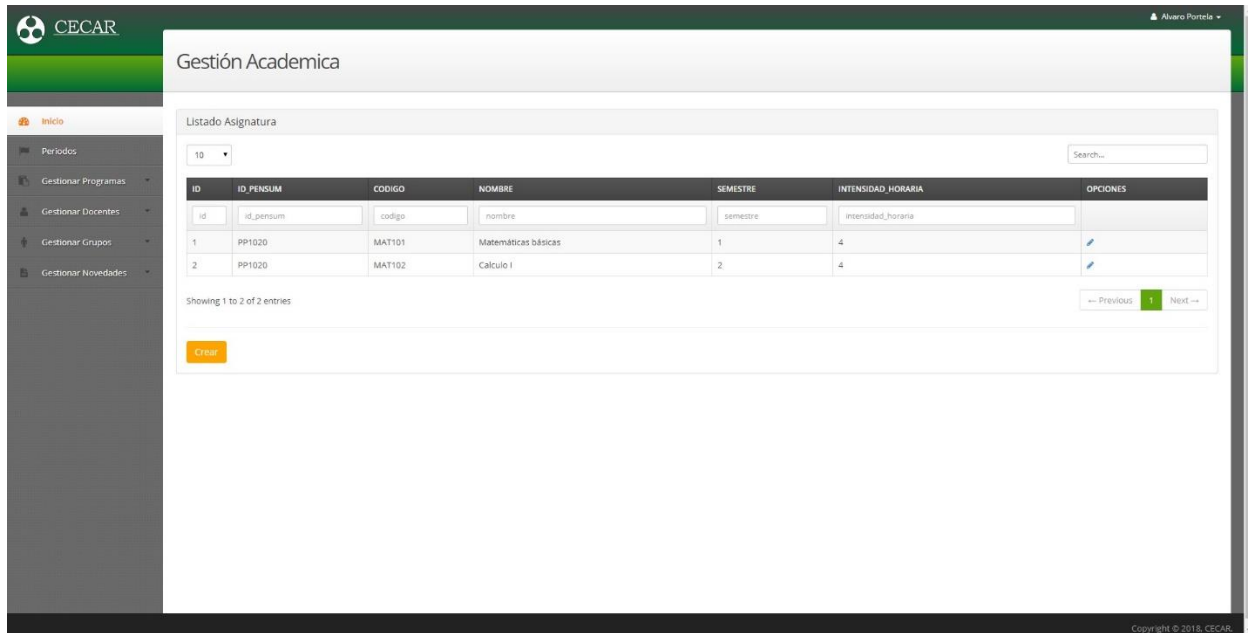


Figura 8. Vista donde se listan las asignaturas de un pensum.

Tabla 17.

Pruebas de historia de usuario #2.

Pruebas de historia de usuario #2.
Tipo de prueba: Caja negra
<p>Prueba #1</p> <ol style="list-style-type: none"> 1. Se seleccionó la opción “Crear programa” del apartado “Gestionar programas” y se creó un programa de ejemplo llamado “Administración”. 2. Se seleccionó la opción “Listar programas” del apartado “Gestionar programas” la cual arroja una tabla con el listado de programas registrados incluido el programa de ejemplo registrado. 3. Se creó un pensum y a su vez una asignatura para el mismo usando los botones de navegación programados sin dificultad y arrojando las vistas correctamente.

2.2.5.2. Iteración 2.

Tabla 18.

Planificación de iteración 2.

N° de historia de usuario	N° de tarea	Nombre de la tarea
1	5	Crear docentes
1	6	Listar docentes registrados.
1	7	Editar docentes
Lenguajes Usados		SQL, C#, HTML, JavaScript.
Herramientas Usadas		Visual Studio, SQL Management Studio Express.
Tiempo estimado de desarrollo		1 semanas.

Tabla 19.

Tarea #5, historia #1.

Formato de tarea	
N° de tarea: 5	N° de historia: 1
Nombre de la tarea: Crear docentes	
Tipo: Desarrollo	Puntos estimados: 2
Responsable: Equipo de desarrollo XP	
Descripción:	
<ul style="list-style-type: none"> • Construir y hacer funcional una vista para la creación de los docentes accedida desde el menú principal y que tenga los campos ID, nombre y tipo de contratación. 	

Tabla 20.

Tarea #6, historia #1.

Formato de tarea	
N° de tarea: 6	N° de historia: 1
Nombre de la tarea: Listar docentes registrados.	

Tipo: Desarrollo	Puntos estimados: 2
Responsable: Equipo de desarrollo XP	
Descripción: <ul style="list-style-type: none">• Crear una función que liste los docentes registrados desde un ítem del menú en una en una vista que contenga una tabla donde cada fila represente un programa académico.	

Tabla 21.

Tarea #7, historia #1.

Formato de tarea	
N° de tarea: 7	N° de historia: 1
Nombre de la tarea: Editar Docentes	
Tipo: Desarrollo	Puntos estimados: 2
Responsable: Equipo de desarrollo XP	
Descripción: <ul style="list-style-type: none">• Crear una vista funcional para la edición de los datos de los docentes y enlazarla a la vista del listado de los docentes a través de un botón.	

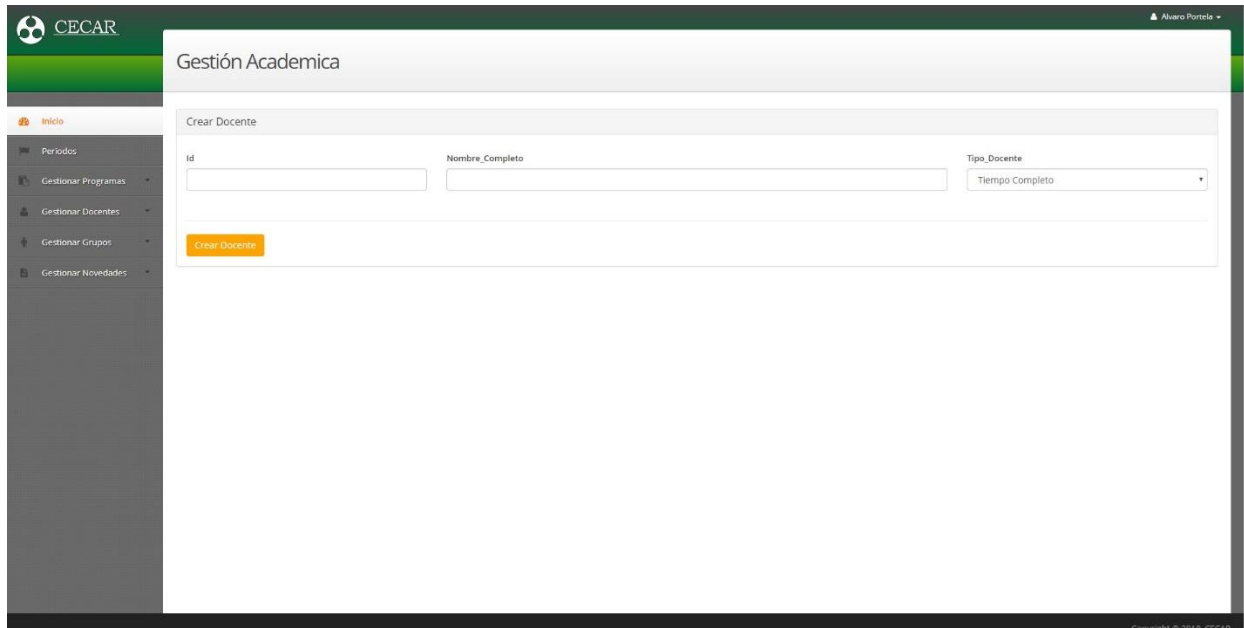


Figura 9. Vista de creación de docentes.

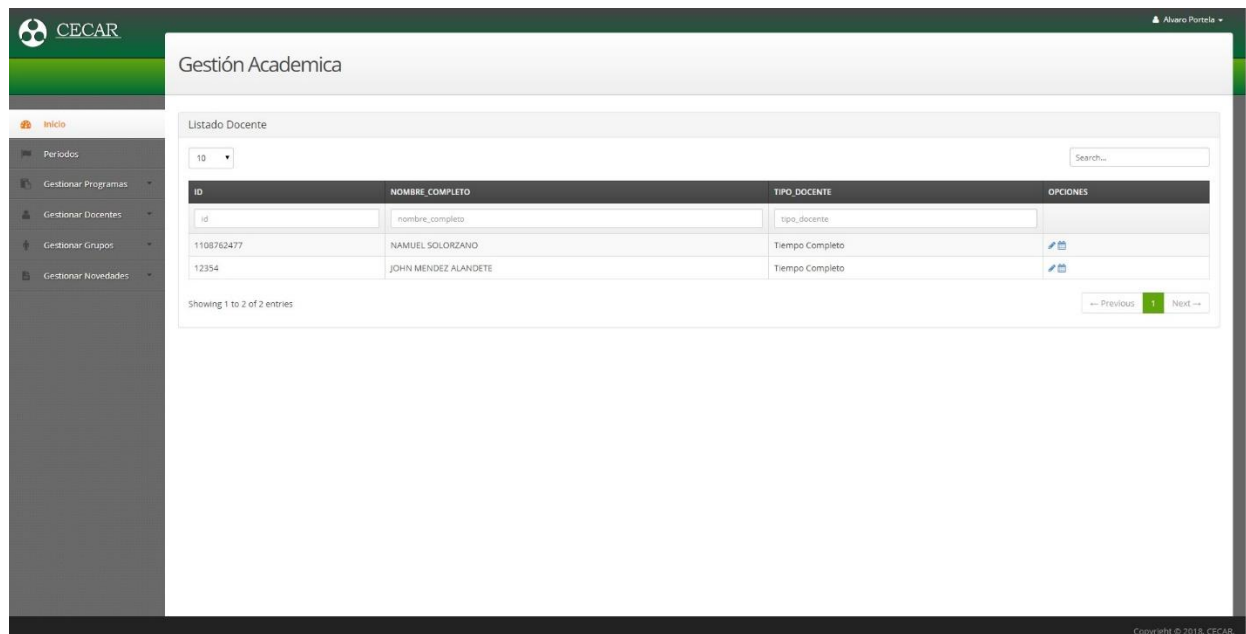


Figura 10. Vista donde se listan los docentes.

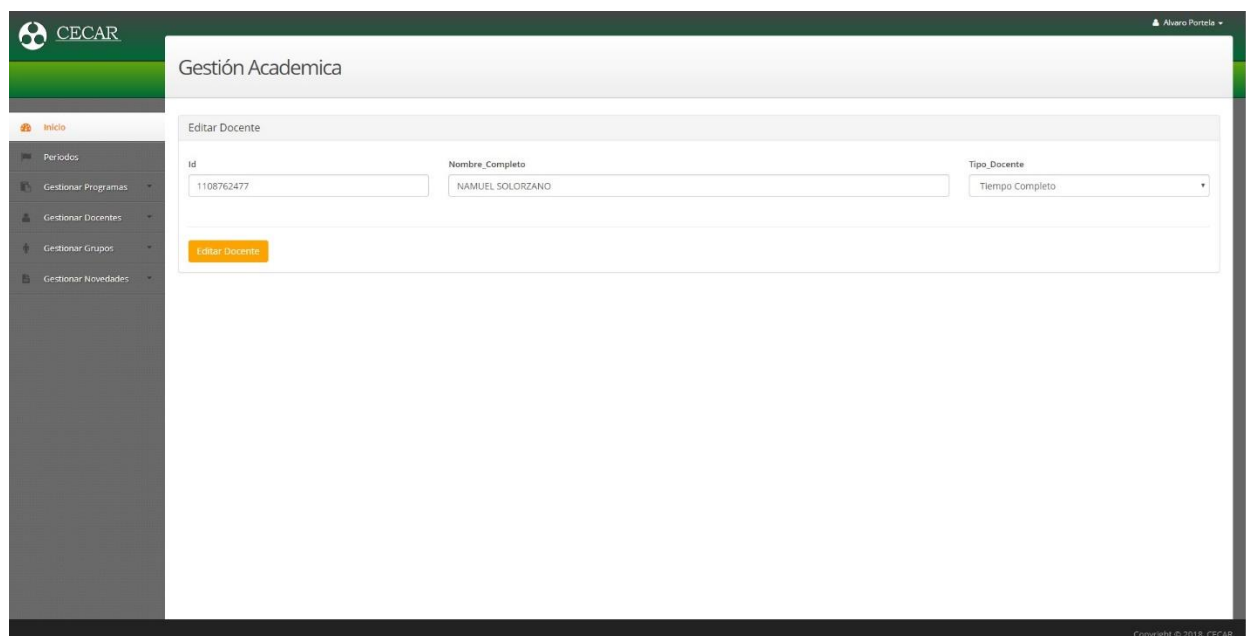


Figura 11. Vista de edición de docentes.

Tabla 22.

Pruebas de historia de usuario #1.

Pruebas de historia de usuario #1.
Tipo de prueba: Caja negra
Prueba #1 <ol style="list-style-type: none"> 1. Se seleccionó la opción crear docente del menú principal y se crearon docentes con sus datos. 2. Se seleccionó la opción de listar los docentes y se obtuvo la lista con los docentes ya registrados. 3. Se editó el nombre de un docente el cual se reflejó en el listado.

2.2.5.3. Iteración 3.

Tabla 23.

Planificación de iteración 3.

N° de historia de usuario	N° de tarea	Nombre de la tarea
3	8	Listar periodos
3	9	Crear periodos
3	10	Editar periodos
Lenguajes Usados		SQL, C#, HTML, JavaScript.
Herramientas Usadas		Visual Studio, SQL Management Studio Express.
Tiempo estimado de desarrollo		1 semanas.

Tabla 24.

Tarea #8, historia #3.

Formato de tarea	
N° de tarea: 8	N° de historia: 3
Nombre de la tarea: Listar periodos	

Tipo: Desarrollo	Puntos estimados: 1
Responsable: Equipo de desarrollo XP	
Descripción:	
<ul style="list-style-type: none"> Diseñar una vista donde se listen los periodos académicos creados y que esta contenga un botón para crear periodos y otro para su edición. 	

Tabla 25.

Tarea #9, historia #3.

Formato de tarea	
N° de tarea: 9	N° de historia: 3
Nombre de la tarea: Crear periodos	
Tipo: Desarrollo	Puntos estimados: 1
Responsable: Equipo de desarrollo XP	
Descripción:	
<ul style="list-style-type: none"> Construir una vista que se enlace con el botón de creación de la vista “listar” que permita crear periodos con una caja de texto que capture su nombre. 	

Tabla 26.

Tarea #10, historia #3.

Formato de tarea	
N° de tarea: 10	N° de historia: 3
Nombre de la tarea: Editar periodos	
Tipo: Desarrollo	Puntos estimados: 1
Responsable: Equipo de desarrollo XP	
Descripción:	
<ul style="list-style-type: none"> Construir una vista que se enlace con el botón de edición de el listado de periodos y que permita editar su nombre. 	

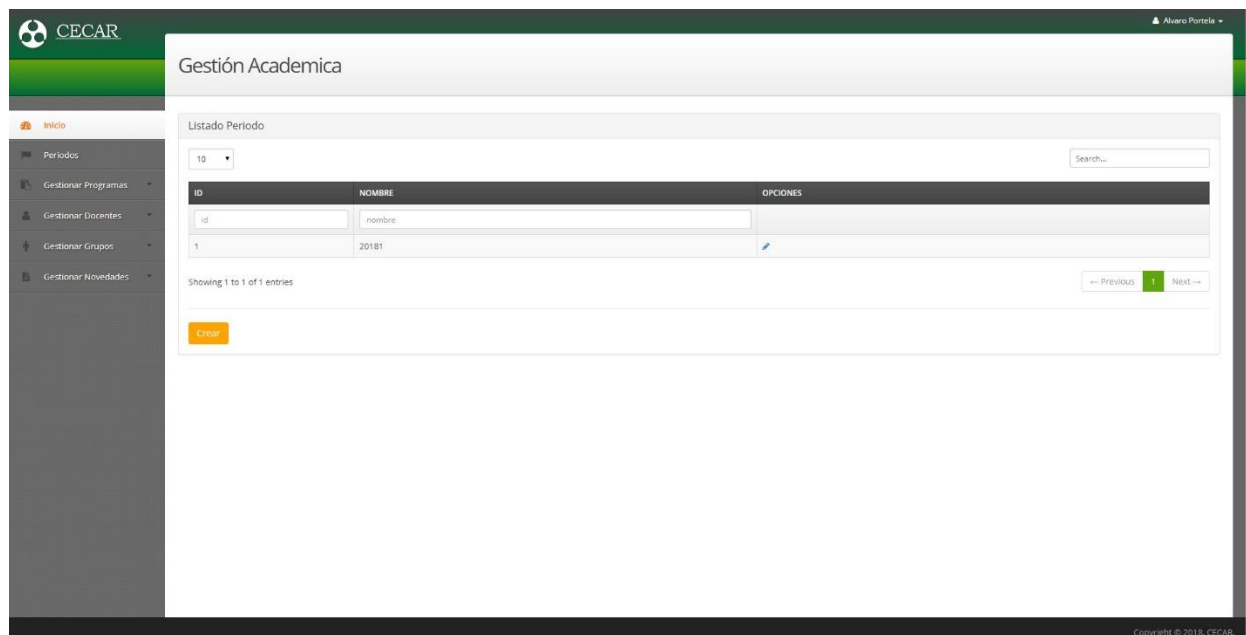


Figura 12. Vista de listado de periodos.

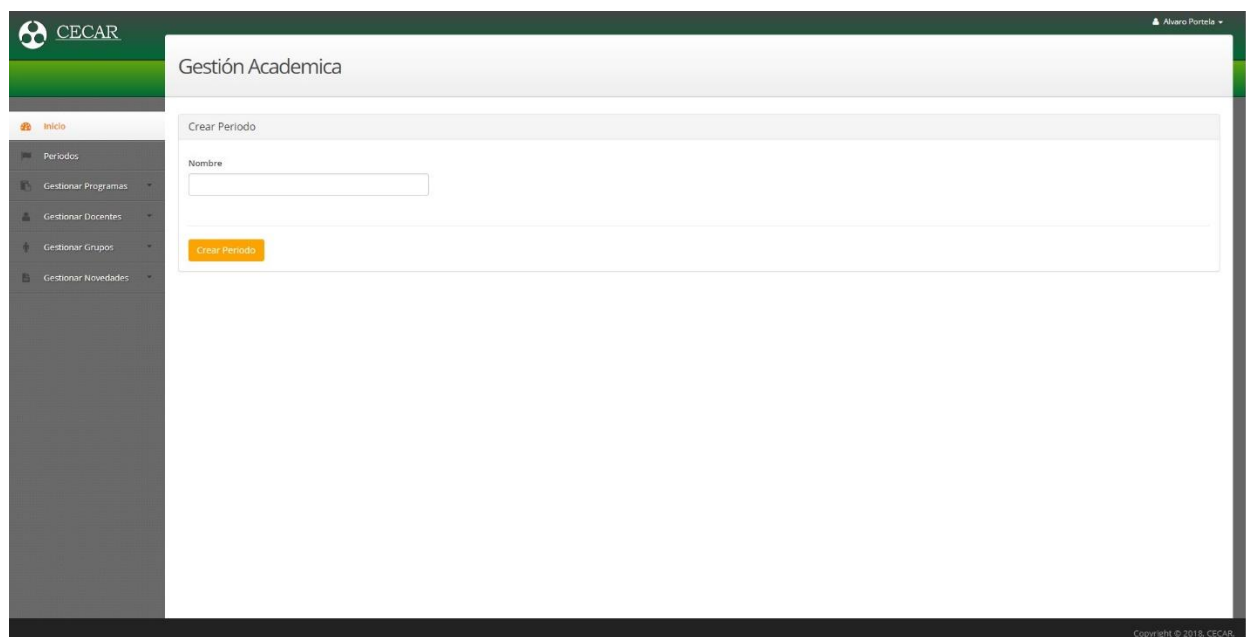


Figura 13. Vista de la creación/edición de periodos.

Tabla 27.

Pruebas de historia de usuario #3.

Pruebas de historia de usuario #3.
Tipo de prueba: Caja negra
Prueba #1 1. Se seleccionó la opción “periodos” del menú principal, inmediatamente se listaron los periodos, acto seguido se creó y edito el nombre de un periodo los tres procesos tuvieron una salida exitosa.

2.2.5.4. Iteración 4.

Tabla 28.

Planificación de iteración 4.

N° de historia de usuario	N° de tarea	Nombre de la tarea
4	11	Crear grupos
4	12	Listar grupos
4	13	Editar grupos
Lenguajes Usados		SQL, C#, HTML, JavaScript.
Herramientas Usadas		Visual Studio, SQL Management Studio Express.
Tiempo estimado de desarrollo		2 semanas.

Tabla 29.

Tarea #11, historia #4.

Formato de tarea	
N° de tarea: 11	N° de historia: 4
Nombre de la tarea: Crear grupos	
Tipo: Desarrollo	Puntos estimados: 3
Responsable: Equipo de desarrollo XP	
Descripción:	

- Se debe diseñar una vista para la creación de los grupos cuya funcionalidad recoja y almacene el pensum, semestre, nombre de la asignatura que se le impartirá a el grupo, periodo, docente, numero de grupo, cantidad de estudiantes y programa de acuerdo al coordinador que se encuentre usando el prototipo.

Tabla 30.

Tarea #12, historia #4.

Formato de tarea	
N° de tarea: 12	N° de historia: 4
Nombre de la tarea: Listar grupos	
Tipo: Desarrollo	Puntos estimados: 3
Responsable: Equipo de desarrollo XP	
Descripción:	
<ul style="list-style-type: none"> • Se debe diseñar una vista que contenga una tabla donde se listen los diferentes grupos ya registrados. 	

Tabla 31.

Tarea #13, historia #4.

Formato de tarea	
N° de tarea: 13	N° de historia: 4
Nombre de la tarea: Editar grupos	
Tipo: Desarrollo	Puntos estimados: 3
Responsable: Equipo de desarrollo XP	
Descripción:	
<ul style="list-style-type: none"> • Construir una vista donde se permita editar los campos de un grupo seleccionado y enlazarla con la vista del listado de los grupos. 	

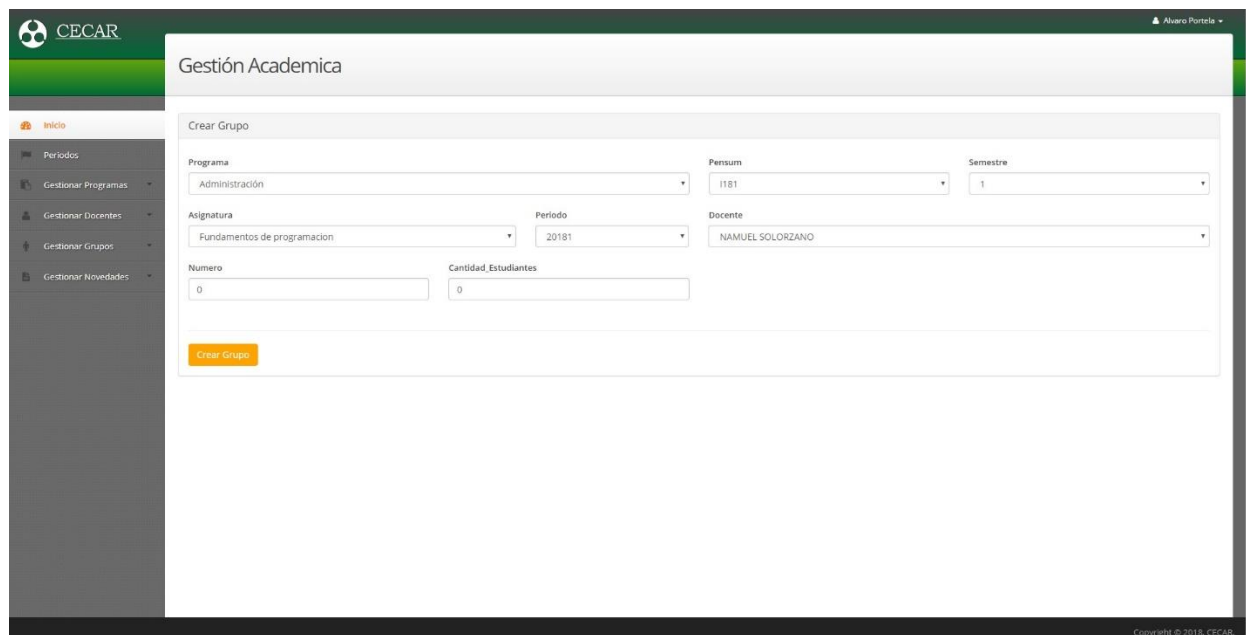


Figura 14. Vista de la creación de grupos.

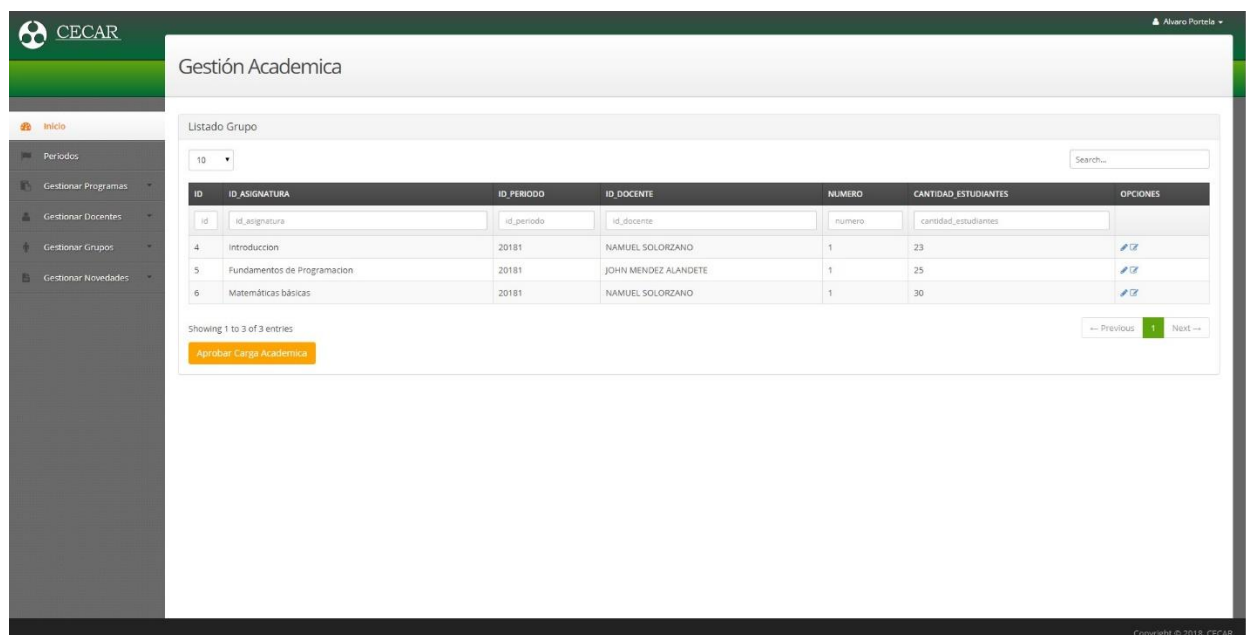
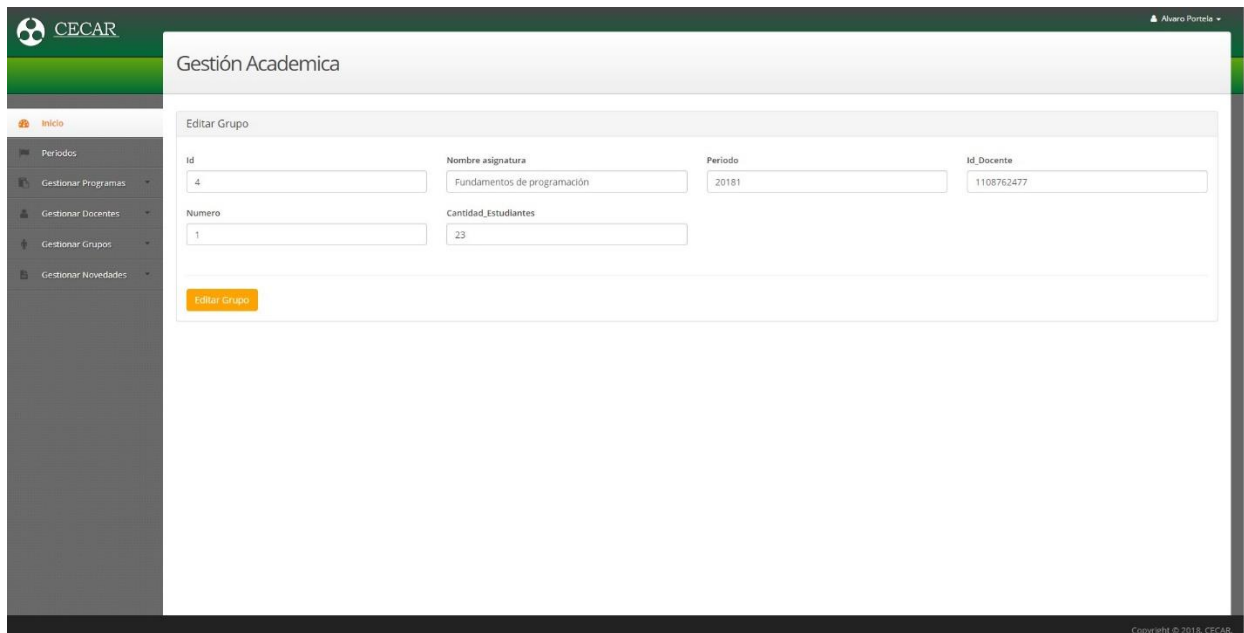


Figura 15. Vista donde se listan los grupos.



CECAR

Inicio

Períodos

Gestionar Programas

Gestionar Docentes

Gestionar Grupos

Gestionar Novedades

Alvaro Portelo

Gestión Académica

Editar Grupo

Id: 4

Nombre asignatura: Fundamentos de programación

Periodo: 20181

Id_Docente: 1108762477

Numero: 1

Cantidad_Estudiantes: 23

Editar Grupo

Copyright © 2018, CECAR

Figura 16. Vista para la edición los grupos.

Tabla 32.

Pruebas de historia de usuario #4.

Pruebas de historia de usuario #4.
Tipo de prueba: Caja negra
<p>Prueba #1</p> <ol style="list-style-type: none"> 1. Se eligió la opción “Crear grupo” del apartado “Gestionar grupos” del menú y se crearon grupos para el programa de ingeniería de sistemas en varias materias. 2. Se seleccionó la opción siguiente y se listaron los grupos anteriormente registrados de manera correcta. 3. Se escogió uno de los grupos y se editaron sus datos los cuales se guardaron correctamente.

2.2.5.5. Iteración 5.

Tabla 33.

Planificación de iteración 5.

N° de historia de usuario	N° de tarea	Nombre de la tarea
6	14	Crear novedades
6	15	Listar novedades
Lenguajes Usados		SQL, C#, HTML, JavaScript.
Herramientas Usadas		Visual Studio, SQL Management Studio Express.
Tiempo estimado de desarrollo		1 semanas.

Tabla 34.

Tarea #14, historia #6.

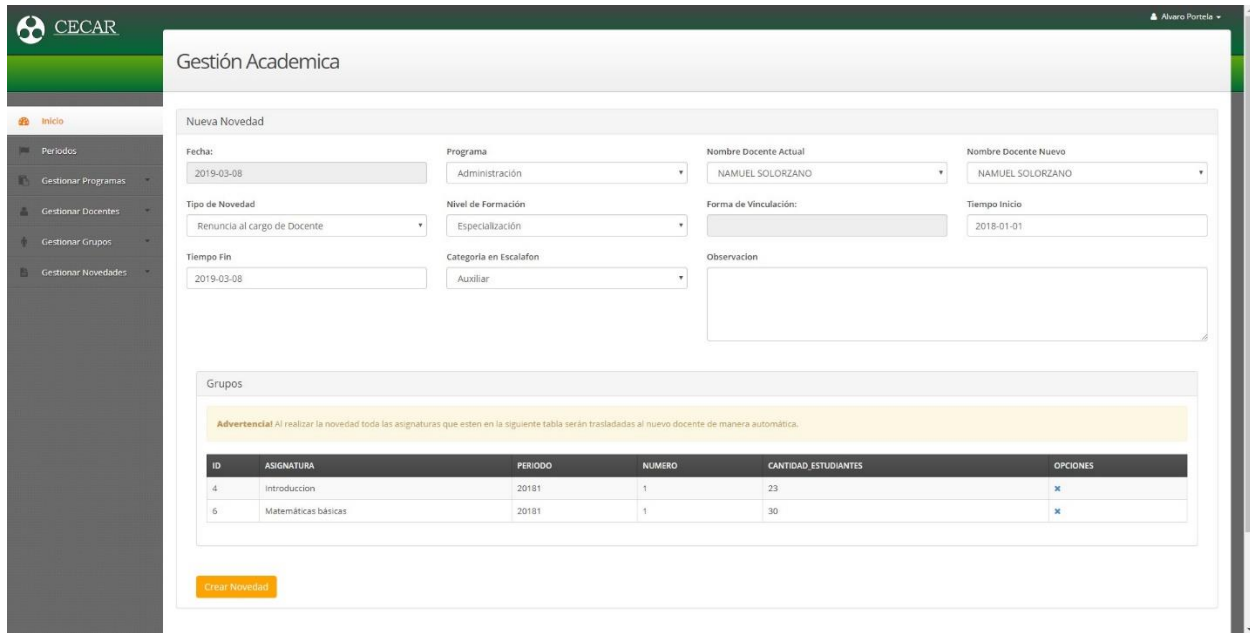
Formato de tarea	
N° de tarea: 14	N° de historia: 6
Nombre de la tarea: Crear novedades	
Tipo: Desarrollo	Puntos estimados: 3
Responsable: Equipo de desarrollo XP	
Descripción:	
<ul style="list-style-type: none"> • Crear una vista que permita registrar las novedades de la carga académica teniendo en cuenta los campos del formato manual de reporte de novedades (Ver anexo B). 	

Tabla 35.

Tarea #15, historia #6.

Formato de tarea	
N° de tarea: 15	N° de historia: 6
Nombre de la tarea: Crear listar novedades	
Tipo: Desarrollo	Puntos estimados: 2
Responsable: Equipo de desarrollo XP	
Descripción:	

- Diseñar una vista donde se listen las novedades de las cargas académicas reportadas, donde se visualicen los campos más relevantes de las mismas.



Nueva Novedad

Fecha: 2019-03-08

Programa: Administración

Nombre Docente Actual: NAMUEL SOLORZANO

Nombre Docente Nuevo: NAMUEL SOLORZANO

Tipo de Novedad: Renuncia al cargo de Docente

Nivel de Formación: Especialización

Forma de Vinculación:

Tiempo Inicio: 2018-01-01

Tiempo Fin: 2019-03-08

Categoría en Escalafón: Auxiliar

Observación:

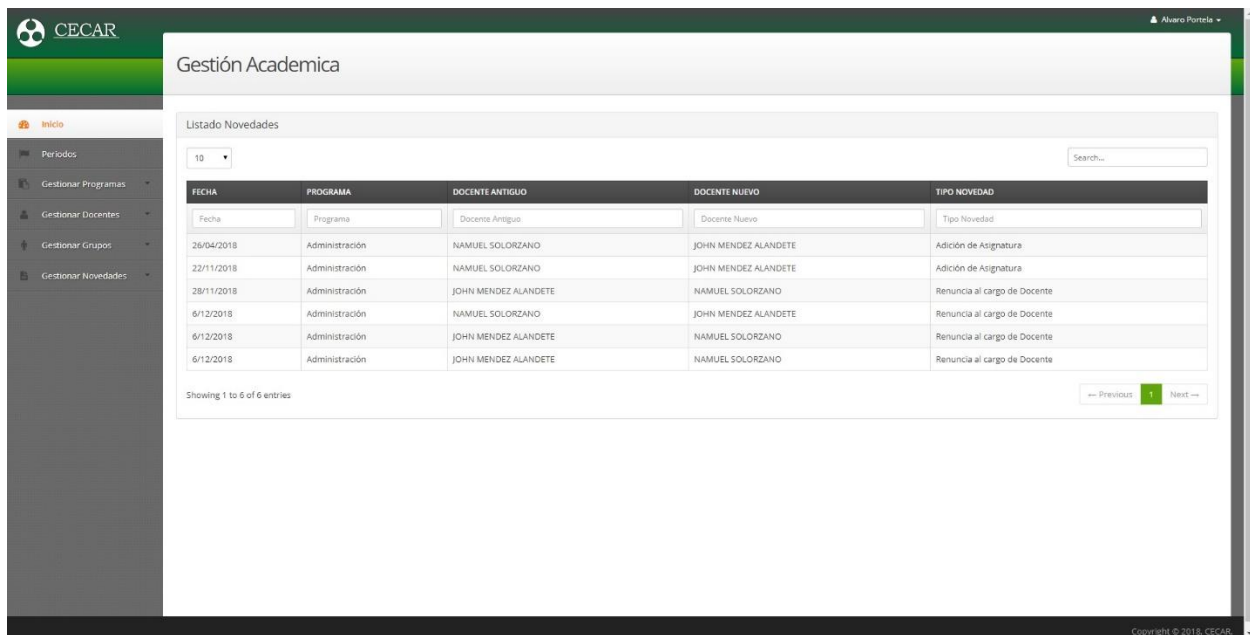
Grupos

Advertencial! Al realizar la novedad toda la asignaturas que estén en la siguiente tabla serán trasladadas al nuevo docente de manera automática.

ID	ASIGNATURA	PERIODO	NUMERO	CANTIDAD ESTUDIANTES	OPCIONES
4	Introducción	20181	1	23	X
6	Matemáticas básicas	20181	1	30	X

Crear Novedad

Figura 17. Vista para la creación de novedades.



Listado Novedades

10

Search...

FECHA	PROGRAMA	DOCENTE ANTIGUO	DOCENTE NUEVO	TIPO NOVEDAD
Fecha	Programa	Docente Antiguo	Docente Nuevo	Tipo Novedad
26/04/2018	Administración	NAMUEL SOLORZANO	JOHN MENDEZ ALANDETE	Adición de Asignatura
22/11/2018	Administración	NAMUEL SOLORZANO	JOHN MENDEZ ALANDETE	Adición de Asignatura
28/11/2018	Administración	JOHN MENDEZ ALANDETE	NAMUEL SOLORZANO	Renuncia al cargo de Docente
6/12/2018	Administración	NAMUEL SOLORZANO	JOHN MENDEZ ALANDETE	Renuncia al cargo de Docente
6/12/2018	Administración	JOHN MENDEZ ALANDETE	NAMUEL SOLORZANO	Renuncia al cargo de Docente
6/12/2018	Administración	JOHN MENDEZ ALANDETE	NAMUEL SOLORZANO	Renuncia al cargo de Docente

Showing 1 to 6 of 6 entries

← Previous 1 Next →

Copyright © 2018, CECAR.

Figura 18. Vista donde se listan las novedades.

Tabla 36.

Pruebas de historia de usuario #6.

Pruebas de historia de usuario #6.
Tipo de prueba: Caja negra
Prueba #1 <ol style="list-style-type: none"> 1. Se creó una novedad usando la opción “crear novedad” del apartado “Gestionar novedades” del menú la cual se creó correctamente y se visualiza un cuadro de confirmación. 2. Se seleccionó la opción “listar novedades” y apareció la novedad registrada anteriormente de manera adecuada.

2.2.5.6. Iteración 6.

Tabla 37.

Planificación de iteración 6.

N° de historia de usuario	N° de tarea	Nombre de la tarea
5	16	Asignar horario
5	17	Comparar y aprobar cargas
5	18	Exportar horarios
Lenguajes Usados		SQL, C#, HTML, JavaScript.
Herramientas Usadas		Visual Studio, SQL Management Studio Express.
Tiempo estimado de desarrollo		1 semanas.

Tabla 38.

Tarea #16, historia #5.

Formato de tarea	
N° de tarea: 16	N° de historia: 5

Nombre de la tarea: Asignar horario	
Tipo: Desarrollo	Puntos estimados: 5
Responsable: Equipo de desarrollo XP	
Descripción:	
<ul style="list-style-type: none"> • Construir una vista que permita por medio de un componente de calendario asignar horas de clase a los diferentes grupos en cada una de las asignaturas, controlando los cruces y la intensidad horaria de las mismas, esta vista debe integrarse a la vista del listado de los grupos a través de un botón. 	

Tabla 39.

Tarea #17, historia #5.

Formato de tarea	
N° de tarea: 17	N° de historia: 5
Nombre de la tarea: Comparar y aprobar cargas	
Tipo: Desarrollo	Puntos estimados: 4
Responsable: Equipo de desarrollo XP	
Descripción:	
<ul style="list-style-type: none"> • Añadir un botón en la vista donde se listan los grupos y que permita aprobar las cargas académicas. • Diseñar una vista donde se muestren la carga académica aprobada y la carga académica actual. 	

Tabla 40.

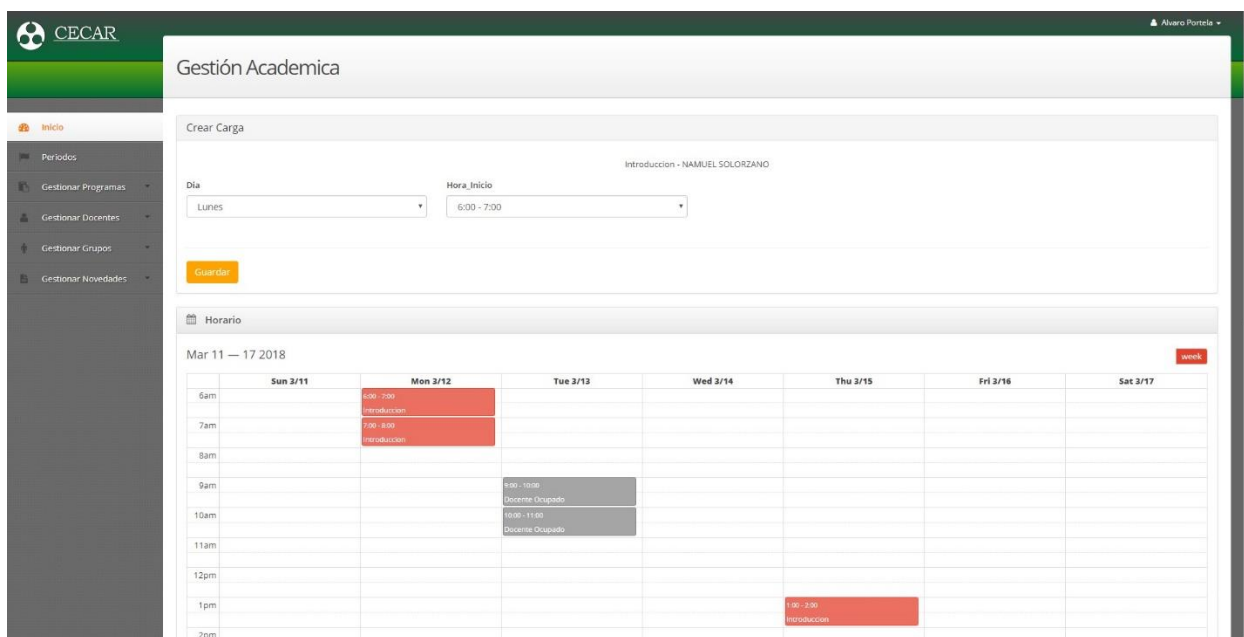
Tarea #18, historia #5.

Formato de tarea	
N° de tarea: 18	N° de historia: 5
Nombre de la tarea: Exportar horario	
Tipo: Desarrollo	Puntos estimados: 4

Responsable: Equipo de desarrollo XP

Descripción:

- Añadir una función que permita exportar los horarios de los diferentes docentes y semestres a formato Excel
- Se debe hacer una vista nueva donde se seleccione el periodo académico del horario a exportar e integrar un botón en la vista donde se listan los docentes.



The screenshot displays the 'Gestión Académica' interface. At the top, there is a 'Crear Carga' form with the following fields:

- Día:** A dropdown menu currently set to 'Lunes'.
- Hora_Inicio:** A dropdown menu currently set to '6:00 - 7:00'.
- Curso:** A text field containing 'Introduccion - NAMUEL SOLORZANO'.
- Guardar:** A yellow button to save the entry.

Below the form is a 'Horario' section showing a weekly grid for the week of March 11 to 17, 2018. The grid has columns for each day and rows for time slots from 6am to 2pm. The following table summarizes the assignments shown in the grid:

Time Slot	Sun 3/11	Mon 3/12	Tue 3/13	Wed 3/14	Thu 3/15	Fri 3/16	Sat 3/17
6am - 7am		6:00 - 7:00 Introduccion					
7am - 8am		7:00 - 8:00 Introduccion					
9am - 10am			9:00 - 10:00 Docente Ocupado				
10am - 11am			10:00 - 11:00 Docente Ocupado				
12pm - 1pm					1:00 - 2:00 Introduccion		
1pm - 2pm							

Figura 19. Componente de asignación de horarios.

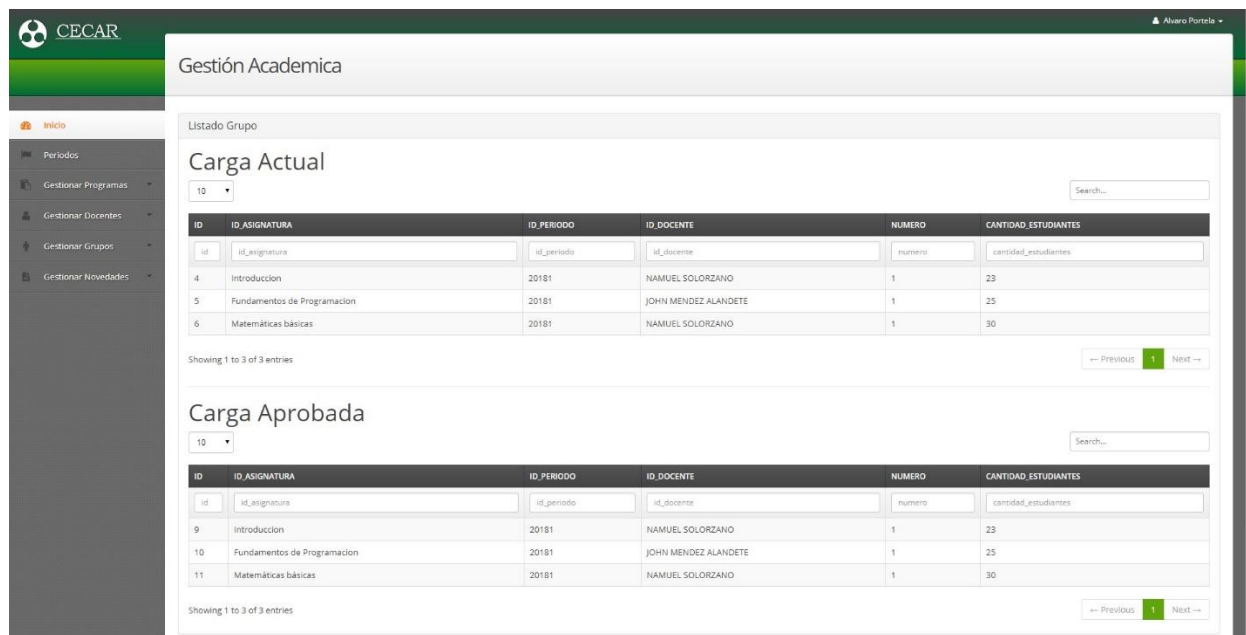


Figura 20. Comparación de carga actual con aprobada.

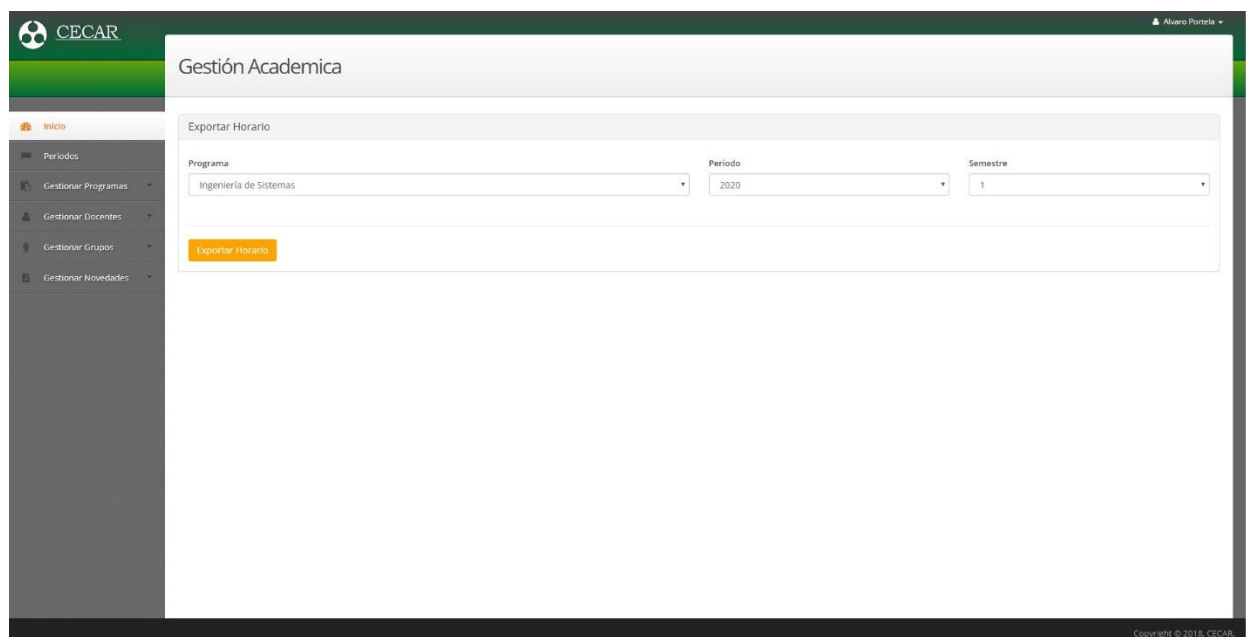


Figura 21. Vista para exportar horario.

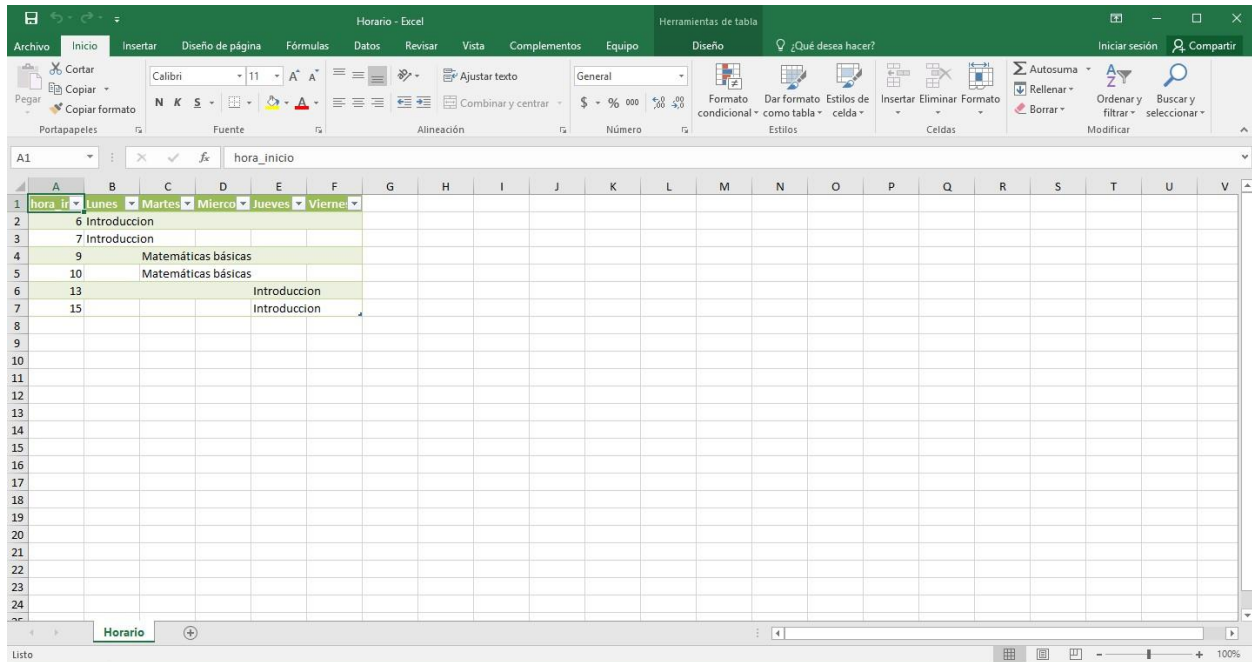


Figura 22. Ejemplo de horario exportado.

Tabla 41.

Pruebas de historia de usuario #5.

Pruebas de historia de usuario #5.
Tipo de prueba: Caja negra
<p>Prueba #1</p> <ol style="list-style-type: none"> 1. Se le asigno horario a algunos grupos creados para las respectivas asignaturas los cuales se mostraron correctamente y se validaron los cruces con cuadros grises. 2. Se dio aprobación por medio del botón respectivo y se mostró correctamente en la vista “comparar cargas”. 3. Se exporto el horario con las asignaturas antes modificadas y correspondió exactamente a lo registrado.

2.3. Fase III: Evaluación del prototipo de software

El prototipo al ser construido de manera iterativa permitió realizar entregas periódicas al coordinador de la facultad de ingeniería de sistemas quien fue tomado de entre la población de coordinadores como sujeto para realizar pruebas de funcionalidad en cada iteración.

Se busca hacer entregas en breves lapsos de tiempo, incrementando pequeñas funcionalidades en cada iteración. Esto conlleva a que el cliente pueda tener una mejor experiencia con el Software, ya que lo que deberá probar como nuevo, será poco, y fácilmente asimilable, pudiendo sugerir mejoras con mayor facilidad de implementación (Bahit 2012, p. 78).

El proceso se llevaba a cabo presentando las funcionalidades añadidas al final de cada iteración al sujeto de pruebas el cual manipulaba el prototipo de software a su antojo y se recibía su aprobación de acuerdo al nivel de cumplimiento de las necesidades del mismo en cada funcionalidad, acto seguido se plasmaba esto como las pruebas de caja negra presentadas en el apartado anterior.

En la última iteración del prototipo se realizó una prueba de funcionalidad general de todas las funcionalidades mínimas esperadas por el prototipo y una vez se constató que eran cumplidas se dio por terminado el proceso de testing con el usuario dando como resultado un prototipo funcional y listo para seguir siendo escalado en un futuro.

Capítulo III

Conclusiones

Día a día la tecnología nos permite cambiar la forma de presentar la información y se ve la necesidad de que las universidades implementen estas tecnologías web que permiten mejorar procesos que se manejan de forma manual permitiendo a su vez incursionar en la red. Con la elaboración de este prototipo se espera servir de apoyo adicional al sistema de autenticación único ALDEA en relación a la planeación preliminar de toda la proyección de grupal y asignación de cargas académicas para todas las facultades de manera organizada, la herramienta final permite la rapidez a la hora de entregar y cargar la información ya que es veraz, segura y verificada; pudiendo dar trámite a las novedades que puedan presentarse de forma rápida y confiable.

Para lograr un desarrollo de software con calidad se debieron utilizar diversas técnicas y buenas practicas, todo esto con la finalidad de seguir un proceso estructurado y ya probado lo que nos lleva a deducir que la implementación de la herramienta software incidirá positivamente en la creación de la carga académica porque mitigara los problemas que se originan a principio de cada semestre detectadas y/o requeridas por el personal encargado a través de una planeación solida sin muchos contratiempos que posibilitara un reporte final más confiable .

Podemos afirmar que la forma manual que se llevaba la planeación de la carga académica año tras año fue fundamental para la eficacia en cuanto a la elaboración de este prototipo de herramienta software se refiere, dado que la madurez en cuanto al proceso proveyó de un punto de partida claro y conciso haciendo que la definición de los requerimientos del proyecto fuese sencilla derivando esto a su vez en una puesta en marcha de las demás fases del proceso más acertada y solida

Basados en las pruebas realizadas al prototipo de software nos permite decir que cumple con los requerimientos inicialmente pensados en gran medida y que la consecución de los objetivos planteados al inicio del proyecto se ha logrado satisfactoriamente.

Recomendaciones

El desarrollo de una herramienta software es un proceso meticuloso y de crecimiento razón por la cual este prototipo de software debe ser tomado como punto de partida para la mejora de los procesos dentro de las coordinaciones académicas de la corporación y apoyara de manera significativa a las herramientas de software pre existentes, por lo tanto, se recomienda continuar el proceso de mejora continua de este, añadiendo más funcionalidad, con el fin de llegar a una implementación precisa dentro del proceso de construcción de cargas académicas.

Dentro de todo proyecto de desarrollo de software es totalmente necesario mantener una relación activa con todos los actores implicados en especial los usuarios finales quienes son los que al final pueden dar una retroalimentación más exacta de acuerdo con su experiencia con el artefacto software por lo tanto es de vital importancia que los encargados de continuar con el proceso de mejora del prototipo se mantengan trabajando de la mano con los coordinadores de los programas académicos de la corporación, permitiendo así no salir de la senda que se ha empezado a seguir para abarcar necesidades pendientes o funcionalidades adicionales.

Puesto que este prototipo se realizó de manera local no puede concebirse que se exploten todas sus capacidades de esta manera un excelente paso a seguir seria lograr su integración total con el sistema de directorio activo de la corporación así como también con todos aquellos aplicativos que así lo requieran esto con el fin de aumentar el control del proceso de manera más precisa, como también se podría implementar un método de alimentación de información más eficiente, robusto y masivo a sabiendas que esta información es cuantiosa y puede resultar incómodo su manejo.

Referencias bibliográficas

- Bahit, E. (2012). Scrum y eXtreme Programming para Programadores. Recuperado de <https://www.safecreative.org/>
- Contreras, J. (2013). *El sistema de registro de la carga académica mediante entornos web. Una propuesta tecnológica para la gestión en la Universidad Nacional Experimental Táchira (Venezuela)* (Tesis doctoral). Recuperado de <https://www.tdx.cat/handle/10803/128183>
- Mejía, J. y Paternina, C. (2010). Asignación de horarios de clases Universitarias mediante algoritmos Evolutivos. *Revista educación en ingeniería*. Recuperado de http://www.acofi.edu.co/revista/Revista9/2010_I_22.pdf
- Pressman, R. (2010). *Ingeniería del software un enfoque práctico* (7ma. ed.). Ciudad de México, México: McGraw-Hill.
- Rincón, H., Rodríguez, J. y Ortégón, P. (2008). *Implantación de un módulo de asignación de horarios para los docentes de ingeniería de la universidad de San Buenaventura sede Bogotá* (Trabajo de grado). Recuperado de <http://biblioteca.usbbog.edu.co:8080/Biblioteca/BDigital/42122.pdf>
- Rivadeneira, S. (2012). Metodologías ágiles enfocadas al modelado de Requerimientos. *Revista de Informes Científicos y Técnicos de la Universidad Nacional de la Patagonia Austral*. Recuperado de <http://secyt.unpa.edu.ar/journal/index.php/ICTUNPA/index>

Anexos

Anexo A. Ejemplo de carga académica real


CORPORACION UNIVERSITARIA DEL CARIBE CECAR															
VICERRECTORIA ACADEMICA															
FORMATO UNICO PARA PRESENTACION DE CARGAS Y MODIFICACIONES ACADEMICAS															
MODALIDAD PRESENCIAL															
PERÍODO ACADÉMICO: I															
FACULTAD: INGENIERIA -ING. SISTEMAS															
AÑO: 2014															
DOCENTES DE TIEMPO COMPLETO Y MEDIO TIEMPO															
1. ÍTEM	2. NOMBRES COMPLETOS (ORDENADO ALFABETICAMENTE)	RESULTADOS EVALUACION DOCENTE	3. TITULO(S) DE PREGRADO Y POSGRADO	4. DEDICACIÓN	5. CATEGORIA EN ESCALAFON DE CECAR	6. PROGRAMA(S)	7. NOMBRE DE LAS ASIGNATURAS ACUERDO CON EL PENSUM Y OTRAS ACTIVIDADES A CARGO	CODIGO ASIG.	8. INF	9. EGR	10. SEMES	11. GRUPO	12. Hrs	13. No. ESTUD.	
1	SANDRA MILENA DE HOYOS C.C. 23179576	2012-1 (7,34)	1	ING. DE SISTEMAS	TC	PROFESOR ASISTENTE	INGENIERÍA DE SISTEMAS	1	LENGUAJES DE PROGRAMACION -II	CCO7010	IV	D	1	4	20
		2012-2 (8,01)	2	ESP. EN DOCENCIA				2	BASES DE DATOS	CCO4005	V	D	1	4	18
		2013-1 (7,59)						3	ACTUALIZACIÓN TECNOLÓGICA	EPF1030	X	D	1	3	12
		2013-2 (8,11)						4	DOCENTE CON ACTIVIDADES INVESTIGATIVAS						29
TOTAL HORAS													40		
2	HERNANDEZ HERNANDEZ GUILLERMO CARLOS C.C.92529908	2012-1 (9,08)	1	ING. SISTEMAS	TC	PROFESOR ASISTENTE	INGENIERÍA DE SISTEMAS	1	AUTÓMATAS Y COMPILADORES	CCO6010	VI	D	1	3	23
		2012-2 (8,23)	2	MG. EN INFORMATICA				2	ANÁLISIS DE ALGORITMOS	CCO6005	V	D	1	3	14
		2013-1 (8,75)	3	CAND. MAGISTER EN EDUCACIÓN				5	ELECTIVA PROFESIONAL II	EPF-1010	X	D	1	4	15
		2013-2 (8,87)						4	SISTEMAS OPERATIVOS	CCO1020	VII	D	1	3	18
TOTAL HORAS													40		
3	MENDEZ ALANDETE JHON CC.92.531.165	2012-1 (8,98)	1	ING. DE SISTEMAS	T.C	PROFESOR ASISTENTE	INGENIERÍA DE SISTEMAS	1	SISTEMAS DISTRIBUIDOS	CCO3010	VIII	D	1	4	16
		2012-2 (7,13)	2	ESP. EN REDES DE COMPUTADORES				2	ANÁLISIS Y DISEÑO II	CCO4015	VII	D	1	4	21
		2013-1 (8,91)	3	CERT.INTERNACION. EN ORACLE				3	TOPICOS ESPECIALES DE ING. DE SISTEMAS II	EPF1025	IX	D	1	4	21
		2013-2 (8,92)	4	MAESTRIA "INGENIERIA DEL SOFTWARE" (en curso)				4	DOCENTE CON ACTIVIDADES INVESTIGATIVAS						28
TOTAL HORAS													40		
4	SOLORZANO PERTALTA NAMUEL FRANCISCO CC.92.555.606	2012-2 (8,14)	1	ING. DE SISTEMAS	T.C	PROFESOR ASISTENTE	INGENIERÍA DE SISTEMAS	1	FUNDAMENTOS DE PROGRAMACIÓN II	CCO7010	II	D	1	4	25
		2013-1 (8,69)	2	ESP. PEDAGIA. PARA EL DLLLO. DEL APZE. AUTONOMO				2	INGENIERIA DEL SOFTWARE	CCO3005	VIII	D	1	4	16
		2013-2 (8,96)	3	ESP. ING. DEL SOFTWARE				3	ANÁLISIS Y DISEÑO I	CCO4010	VI	D	1	4	13
			4	DEL SOFTWARE IV SEM(EN CURSO)				4	ELECTIVA PROFESIONAL-I	EPF1005	IX	D	1	4	20
								5	ESTRUCTURA DE DATOS II	CCO7020	IV	D	1	4	20
TOTAL HORAS													40		

Anexo B. Formato de reporte de novedades

CORPORACIÓN UNIVERSITARIA DEL CARIBE - CECAR
VICERRECTORÍA ACADÉMICA

Personería jurídica No. 7786 MEN - ICFCB - No. 892.201.263-1

FORMATO PARA REPORTE DE NOVEDADES PERSONAL DOCENTE



CECAR
Corporación Universitaria del Caribe

FECHA:

FACULTAD: _____

NOMBRE Y APELLIDOS DEL DOCENTE: _____
CÉDULA DE CIUDADANÍA No. _____

NOMBRE Y APELLIDOS DEL DOCENTE REEMPLAZADO: _____

TIPO DE NOVEDAD QUE SE REPORTA	TIEMPO CONTRATADO											
	D			M			A					
renuncia) • Adición de Asignatura • Adición de Actividad • Supresión de Asignatura • Supresión de Actividad • Renuncia en Asignatura • Renuncia a Actividad • Nombramiento Nuevo Docente (anexar hoja de vida) • Otro	Fecha de Inicio:						Fecha de Finalización:					
	NIVEL DE FORMACIÓN											
	Doctorado		Maestría		Especialización		Diplomado en Docencia					
							CECAR		Otra IES			
	CATEGORÍA EN ESCALAFÓN											
	Auxiliar		Asistente		Asociado		Titular					
* CUAL?	<input style="width: 100%;" type="text"/>											


FORMA DE VINCULACIÓN	
Tiempo Completo	<input type="checkbox"/>
Medio Tiempo	<input type="checkbox"/>
Catedrático	<input type="checkbox"/>

TIPO DE CONTRATO	
Termino Fijo	<input type="checkbox"/>
Labor Contratada	<input type="checkbox"/>

ESPECIFICACIÓN DE ASIGNATURA / ACTIVIDAD							
PROGRAMA	ASIGNATURA	CODIGO	SEMESTRE	JORNADA	GRUPO	HORA	N. EST

OBSERVACIONES:

FIRMA DECANO



CECAR
Corporación Universitaria del Caribe