



Desarrollo de software de gestión para la carga horaria semestral de asignaturas y/o módulos de una carrera universitaria

Autor: Fabio Durán Verdugo

Consultor: Gregorio Robles Martinez

Tutor externo: Gabriel Nuñez Vivanco

Especialidad: Desarrollo de Software

Fecha: 8 de Enero de 2017

Licencia de la publicación:

Este proyecto se publica bajo la licencia GFDL.

Usted es libre de:

- **Copiar, distribuir y comunicar públicamente la obra**
- **Re-mezclar -transformar la obra**
- **Hacer un uso comercial de esta obra**

Bajo las condiciones siguientes:

- **Reconocimiento** - Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciadador (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).
- **Compartir bajo la misma licencia** - Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

Entendiendo que:

- **Renuncia** - Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor
- **Dominio Público** - Cuando la obra o alguno de sus elementos se halle en el dominio público según la ley vigente aplicable, esta situación no quedará afectada por la licencia.
- **Otros derechos** - Los derechos siguientes no quedan afectados por la licencia de ninguna manera:
 - Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior.
 - Los derechos morales del autor.
 - Derechos que pueden ostentar otras personas sobre la propia obra o su uso, como por ejemplo derechos de imagen o de privacidad.
- **Aviso** - Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.

Resumen del Proyecto

En el presente informe se describe la construcción de un software de libre acceso, para una problemática real basada en la generación y asignación de horarios para módulo de la carrera Ingeniería Civil en Bioinformática de la Universidad de Talca, para periodos de tiempo en particular.

Basado en la problemática y el interés de mejorar los tiempos y otorgar un mejora, se generó una herramienta que pueda optimizar y organizar la asignación de horarios a diversos cursos que pertenecen a una carrera determinada, dependiendo del tipo de asignatura, nivel semestre, el profesor y entre otras características particulares que puedan resultar.

Indice

1. Introducción:	8
1.1 Contexto:	8
1.1.1 La problemática:	8
1.1.2 Escuela de Ingeniería en Bioinformática:	9
1.1.3 Universidad de Talca:	10
1.2 Situación Actual escuela de Ingeniería Civil en Bioinformática:	10
1.3 Motivación:	11
1.4 Tecnología empleadas:	12
1.4.1 Django:	12
1.4.2 PostgreSQL:	13
1.4.3 Bootstrap:	14
1.4.4 Git:	15
1.4.4.1 Github:	16
2. Objetivos y planificación:	19
2.1 Objetivos:	19
2.1.1 Objetivo general:	19
2.1.2 Objetivo específicos:	19
2.2 Plan de trabajo:	19
2.2.1 eXtreme Programming:	20
2.2.2 Scrum:	20
2.3 Metodología del Trabajo:	21
2.4 Carta Gantt:	23
2.5 Equipo de Trabajo:	23
3. Requerimientos y solución:	26
3.1 Requerimientos:	26
3.1.1 Concepto de historias de usuarios:	26
3.2 Definición de historias de usuarios para el proyecto:	27
3.3 Relatos de historias de usuario:	27
3.3.1 Primera iteración:	28
3.4 Interacción para toma de requerimientos:	39
3.5 Diseño de la solución:	39
3.5.1 Diagrama Entidad Relación ER:	40
3.5.2 Estructura de la aplicación - Diagrama de clases:	40
3.5.3 Estructura de aplicación:	42
3.5.4 Definición de modelo en Django:	43
3.5.5 Definición de formularios:	44
3.5.6 Definición de Vistas en Django:	45
3.5.7 Definición de los Templates:	46
3.6 Evolución de la interfaz de usuario:	48
3.6.1 Aplicación para parámetros:	48
3.6.2 La aplicación Horario:	49
3.7 Extensiones extras de Django utilizadas:	51
3.8 Extensión extras de Bootstrap3:	51
3.9 Repositorio:	51
3.10 Pendientes por resolver:	52
4. Resultados:	54

4.1 Visualización de la aplicación.....	54
5 Conclusiones.....	63
5.1 Cumplimiento del objetivo.....	63
5.2 Conclusiones personales.....	63
5.3 Conocimientos aplicados y aprendizajes.....	63
5.4 El futuro.....	64
6. Referencias.....	66
7.1 – Licencia del Documento publicado.....	68
GNU Free Documentation License.....	68

Capítulo 1

Introducción

1. Introducción:

A continuación se detalla cuales son los antecedentes del proyecto, situación actual y la motivación para su desarrollo. Así mismo, se describen las tecnologías que se van a utilizar..

1.1 Contexto

1.1.1 La problemática

Nace de la necesidad de construir una herramienta que genere una uniformidad en la asignación de horarios a los diversos módulos de una carrera universitaria. La situación requería que se visualizase y permitiese planificar diversas carga académicas para un plan de estudio en un semestre determinado, además de considerar al profesorado, los distintos niveles y los módulos que correspondiesen o se abrieran en un semestre puntual.

Actualmente la Universidad de Talca no cuenta con algún software que permita el desarrollo para la gestión de dicha tarea, por lo que cada director de escuela debe buscar o crear su propio mecanismo (generalmente en planillas de cálculo) para la organización de una carga académica.

Algo que dificulta más aún esta actividad, es la rotación de directores de escuela o encargados de la planificación semestral, obligando casi siempre a la generación de un nuevo modelo para las establecidas por protocolos.

A considerar que cada carrera puede variar su plan de estudio por un sin número de variables, emitiendo una resolución absoluta por parte de la Universidad de Talca, por lo que los estudiantes de distintos niveles (plan de estudio) pueden estar dentro de un mismo módulo. Una resolución entrega directrices obligatorias de como se debe llevar a cabo la ejecución de un plan de estudio y las distintas equivalencias para un programa determinado.

Dentro de la escuela de Ingeniería Civil en Bioinformática, existen dos planes de estudio paralelos y vigentes, con ello dos propuestas de carga horaria diferentes, pudiendo darse la situación de homologar las asignaturas de ambas propuestas, en solo plan de estudios (definido como horas de actividades relacionadas), que deben ser programadas en conjunto,

siempre estableciendo un criterio de acuerdo a la situación del plan de estudio establecido en la resolución emitida.

Para la distribución de una carga horaria se debe tener en cuenta que una asignatura tiene una cantidad de horas y estos corresponden a un trabajo presencial y autónomo que conlleva a actividades para el estudiantes tales como:

- Clases magistrales
- Seminarios
- Ayudantías
- Laboratorios
- Taller
- Trabajo autónomo.

Cada una de estas actividades a excepción del trabajo autónomo se deben programar y homologar según sea el caso para los diversos planes de estudios.

1.1.2 Escuela de Ingeniería en Bioinformática

Ingeniería Civil en Bioinformática [1] es una carrera multidisciplinaria en donde confluyen las Ciencias de la Computación, Biología, Matemáticas y Administración, con el propósito de colaborar y liderar equipos de trabajo para organizar, analizar, modelar, simular, visualizar y hacer predicciones sobre datos tanto de origen biológicos como de diversa índole.

La carrera de Ingeniería en Bioinformática nace el año 2003 en la Universidad de Talca, siendo la primera en su tipo en Chile y Latinoamérica, impulsada por los nuevos desafíos de la ciencia y la creciente necesidad de contar con profesionales que sean capaces de comprender fenómenos y procesos biológicos, abordando su estudio a través del desarrollo de herramientas informáticas. Producto del Mejoramiento continuo de los procesos formativos y el Convenio de Desempeño de Armonización Curricular, se evoluciona a la Ingeniería Civil en Bioinformática.

1.1.3 Universidad de Talca

La Universidad de Talca [2] es una institución estatal chilena, con sede en la ciudad de Talca, Región del Maule. Es miembro de la Agrupación de Universidades Regionales de Chile, y una de las dieciséis universidades del Consorcio de Universidades del Estado de Chile, la cual pertenece al Consejo de Rectores de las Universidades Chilenas. Su Casa Central se encuentra en Talca, Chile. capital de la VII Región, distante a 250 kilómetros al sur de Santiago.

Posee cerca de siete mil alumnos en 24 carreras de pregrado. Cuenta con 25 programas de postgrado: 20 Magísteres, 5 doctorados y realiza 4 programas de especialización. Cuenta con un Campus en la localidad de Los Niches, cerca de Curicó, donde se encuentra la Facultad de Ingeniería. En la ciudad de Santiago posee dos campus, uno en la comuna de Providencia y otro en San Joaquín. En la ciudad de Linares posee una sede provisional desde 2014, proyectándose construir un campus en 2015.2 También posee un instituto profesional, el Instituto Tecnológico Colchagua.

La Universidad de Talca figura como la onceava universidad chilena según la clasificación webométrica del CSIC, cuarta a nivel nacional según ranking mercurio 2013, primer lugar en calidad académica ranking revista Qué Pasa y en el noveno lugar del ránking QS World University. En el ranking 2011 del QS World University Ranking se ubicó en el puesto 71 a nivel latinoamericano y novena a nivel nacional. Además se encuentra en el primer lugar del ranking fundación Pro Acceso sobre transparencia universitaria con un porcentaje de cumplimiento del 94%

1.2 Situación Actual escuela de Ingeniería Civil en Bioinformática

En la actualidad la escuela de Ingeniería Civil en Bioinformática posee un método de planificación de carga académica manual mediante una planilla de cálculo. Esta es manipulada y modificada dependiendo de la cantidad de módulos para programar, además año a año se va adecuando a las formas de uso dependiendo de los usuarios que manipulen dicha planilla.

Actualmente se utiliza un fichero (imagen 1.2) como herramienta para mantener una estructura de organización, pero en manifestación de los usuarios, esta posee varias problemáticas, un ejemplo es

que a la hora de generar validaciones para situaciones puntuales, estas no pueden ser ejecutadas de forma automática, siendo esto un factor negativo además si consideramos que actualmente una persona puede tomar días en coordinar un horario para módulos del mismo nivel y profesor.

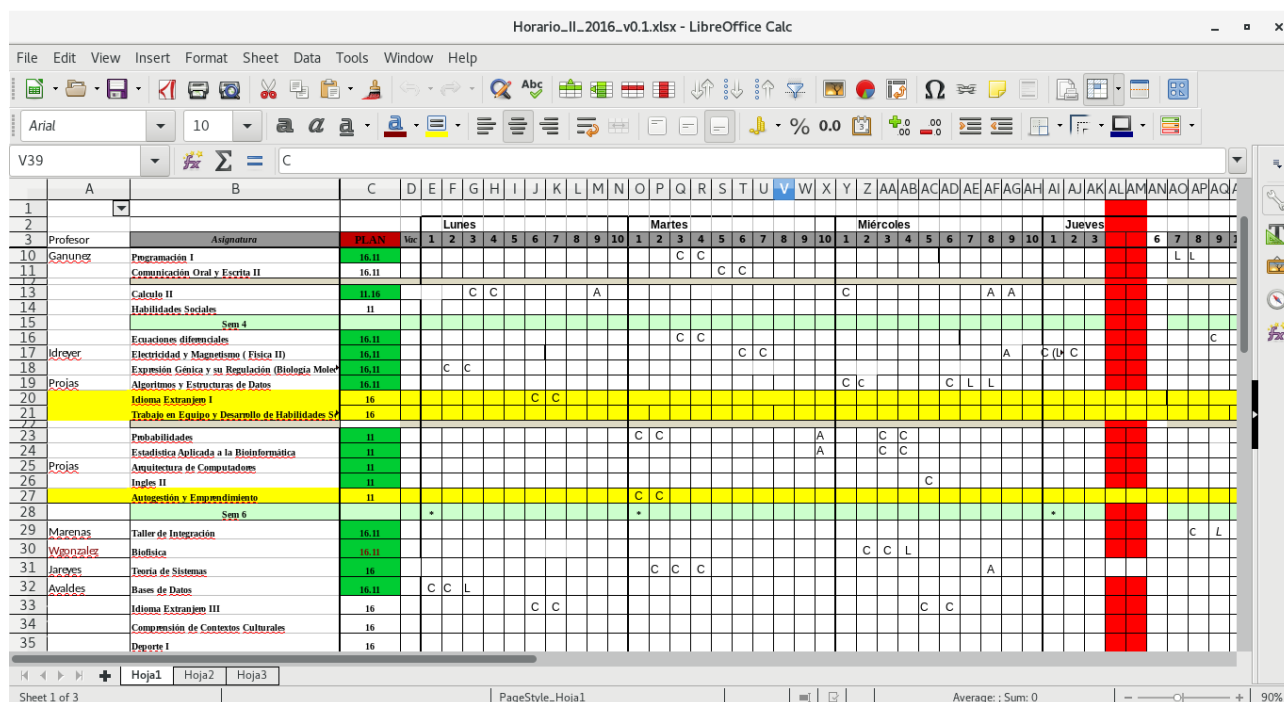


Imagen 1.2 – Planilla de calculo usada actualmente para planificación de módulos

1.3 Motivación

La motivación principal es dar una solución para un problema particular y real, generando una herramienta libre que aporte en la realización de actividades y procesos de organización administrativa que se encomiendan en la carrera de Ingeniería Civil en Bioinformática.

Otro aspecto importante de señalar, es que al existir la carencia de un software que realice la actividad de gestión de carga académica, abre la posibilidad de crear comunidad y que otras personas de otros departamentos o carreras de la universidad aporten con nuevas características y requerimientos para el software, logrando un producto integral para toda la institución.

1.4 Tecnología empleadas:

Acerca de las herramientas generales utilizadas para el desarrollo:

1.4.1 Django

Django según su sitio oficial [3] “Es un framework web de alto nivel en Python que fomenta el desarrollo rápido y el diseño limpio y pragmático, logrando con esto la creación de mejores aplicaciones Web de manera más rápida y con menos código.

Está diseñado para que sea escrito en el lenguaje Python siendo la razón para definirse como un framework de alto nivel, correspondiendo a esto como consecuencia para que Django logre desarrollarse y desenvolverse con todas las características y potenciales que un lenguaje de programación alto nivel puede ofrecer.

Algunas particularidades de Python se centran en escribir código fácil de entender, indentación forzada, desarrollo de aplicaciones muy rápidas y potentes. Además de esto, Python facilita el desarrollo rápido, una de las definiciones de Django, esto debido a que Python es un lenguaje interpretado y con tipado dinámico de una concisa pero expresiva sintaxis que lo hace sencillo de interpretar.

Dentro de sus características se encuentran:

- Entrega una manera para designar qué código se ejecutará para cada URL e intuitiva de corroborar
- Facilita mostrar, validar y volver a mostrar formularios HTML.
- Tiene la habilidad de convertir los datos de un formulario HTML en objetos, tipos de datos o estructuras nativos como arreglos o diccionarios para Python, facilitando la manipulación de los mismos.
- Separar el contenido de la presentación mediante un sistema de plantillas, de manera que se pueda cambiar el aspecto de un sitio web sin afectar al contenido, y viceversa.

- Se integra cómodamente con las capas de almacenamiento (clase Model) logrando realizar CRUD (Create, Read, Update, Delete / Crear, Leer, Actualizar y Borrar) de modo más simple.
- Se puede activar un sistema de administración activo listo para ser utilizado sin ningún tipo de configuración.

Django desarrolla el software para que el código acceda de un modo o en una capa a los datos o modelos, separando la lógica de negocio y a su vez como ya se ha mencionado separa la interfaz de usuario de los puntos anteriores. A esto se le define como patrón de arquitectura de Modelo, Vista, Controlador (MVC), que Django sigue fielmente.

1.4.2 PostgreSQL:

Es un sistema de base de datos de gran alcance, de código abierto objeto-relacional. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada, ganando una sólida reputación de fiabilidad, integridad de datos y exactitud. Se ejecuta en todos los principales sistemas operativos, incluyendo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows [4].

Alguna de sus características más importantes son:

- Es una base de datos que ofrece al 100 % atomicidad, consistencia, aislamiento, durabilidad (ACID)
- Integridad referencial, o sea que la clave externa de una tabla de referencia siempre debe aludir a una fila válida de la tabla a la que se haga referencia.
- Tablespace, o almacen de ficheros con estructura lógica.
- Mecanismo de transacciones anidadas o save points.
- PITR - Point in Time Recovery, un tipo de backup avanzado para trabajos con datos importantes los cuales no pueden perderse en caso de fallo.
- Copias de seguridad en caliente (Online/hot backups).
- Tiene la habilidad de crear herencia entre tablas, por lo que se pueden incluir entre gestores objeto-relacionales.
- Multi-Version Concurrency Control (MVCC).

- Múltiples métodos de autenticación.
- Permite la definición de funciones personalizadas en diversos lenguajes como: PL/pgSQL, PL/Tcl, PL/Perl, PL/Python, PL/PHP, PL/Ruby, PL/Java.
- PostgreSQL es distribuido bajo una licencia similar a BSD y MIT, que les permite a los usuarios hacer cualquier cosa que quieran con el código.

El desarrollo de PostgreSQL es realizado por un equipo conformado en su mayoría por desarrolladores voluntarios esparcidos por todo el mundo, que se comunican vía Internet. Es un proyecto de comunidad y no está controlado por ninguna compañía o instancia comercial.

1.4.3 Bootstrap:

Bootstrap es el más popular framework front-end para HTML , CSS y JavaScript para el desarrollo responsive de pantallas móviles y web. Su dirección web para obtener el ambiente es: <https://getbootstrap.com/>

Inicialmente fue creado como una solución interna para Twitter y posteriormente liberado al público en agosto del 2011 como un proyecto OpenSource en GitHub. Gracias a esto y posterior a su liberación, la comunidad apoyó activamente al proyecto en mejorarlo y entregando una completa suite de herramientas que apoya el desarrollo gráfico web para la correcta y atractiva visualización en toda clase de dispositivos, ahorrando trabajo de tener que rediseñar un sitio web en diversos entornos en demanda.

La idea de la versión actual de Bootstrap (imagen 1.4.3) es que primero se diseña para el dispositivo más pequeño y luego ir adaptando para pantallas mayores, se dice que esto logra mejores resultados de acuerdo a las tendencias y auge de dispositivos móviles.

Una de las grandes características es que Bootstrap nos permite mantener en paralelo hasta 4 diseños de la interfaz en función del tamaño de la pantalla, gracias a un juego de clases CSS con las que determinar la distribución en función del dispositivo en el que se esté visualizando.



Imagen 1.4.3 - Ejemplo de distintos tamaños de imágenes para Bootstrap

En general Bootstrap permite:

- Utilizar muchos elementos web, desde iconos a desplegables, combinando HTML5, CSS y Javascript.
- Un diseño adaptable sin importar el dispositivo, la escala o resolución.
- Una completa integración a un sin número de librerías Javascript.
- Integración a diversos framework o implementaciones de desarrollo web como Django, Wordpress, Joomla, Drupal, Symfony, Ruby on Rails, etc.

1.4.4 Git

Git, es un software de control de versiones diseñado por Linus Torvalds, para en primera instancia ser la herramienta que controle y administre los cambios en el repositorio de la comunidad que desarrolla el Kernel de Linux en el año 2005.

Desde esos años las características que mantiene Git son:

- Velocidad.
- Un simple Diseño.
- Fuerte apoyo al desarrollo no lineal (miles de ramas paralelas).
- Totalmente distribuido.
- Capaz de manejar grandes proyectos como el kernel de Linux eficientemente (velocidad y tamaño de datos)

Como se menciona en su página oficial, una de las principales diferencia entre Git y cualquier otro VCS (Subversion y compañía incluidos) es la forma en cómo Git representa sus datos. Conceptualmente, la mayoría de los demás sistemas almacenan la información como una lista de cambios en los archivos. Estos sistemas (CVS, Subversion, Perforce, Bazaar, etc.) modelan la información que almacenan como un conjunto de archivos y las modificaciones hechas sobre cada uno de ellos a lo largo del tiempo.

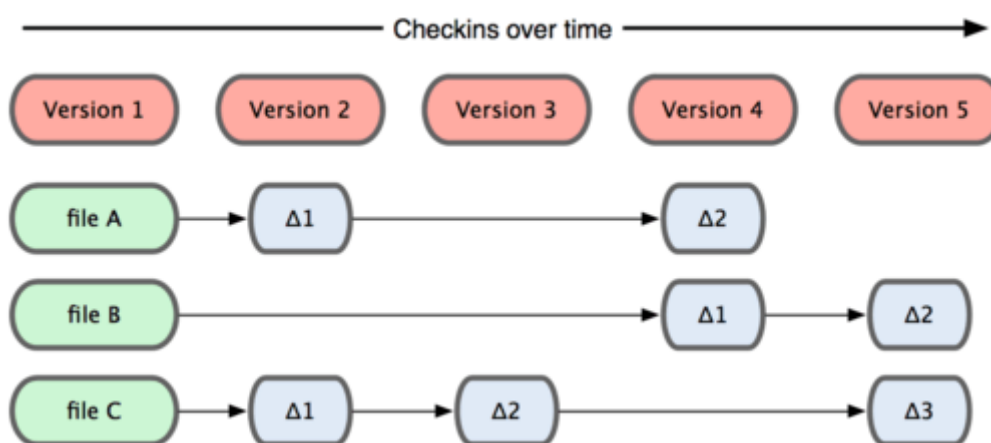


Imagen 1.4.4 - Otros sistemas tienden a almacenar los datos como cambios de cada archivo respecto a una versión base.

Otra diferencia remarcada en la web oficial [5] es la eficiencia con la que se actúa (imagen 1.4.4), puesto a que Git se basa en que cada programador almacena una copia completa del repositorio en su máquina de forma local, incluido el historial de cambios. Esto implica que muchas de las operaciones realizadas sobre el código fuente no tienen lugar en la red, permitiendo que la velocidad de proceso dependa únicamente en los recursos locales.

Git se publica bajo la licencia GNU General Public License versión 2.0.

1.4.4.1 Github

GitHub es una plataforma web de desarrollo colaborativo para alojar proyectos de software utilizando el sistema de control de versiones primario es Git, cuyo objetivo se puede desprender al crecimiento de proyectos de software libre y creación de comunidades alrededor de este.

Además de ser un control de versiones, GitHub entrega una serie de herramientas para que la administración y utilidades para el trabajo en equipo, tales como:

- Revisor de Ramas para un repositorio.
- Un sistema de *bugtracking* o seguimiento de incidencias.
- Herramientas para revisión de código.
- Una wiki para generar documentación relevante para el proyecto.
- Creación de *Fork* o clonación de repositorios de modo instantáneo a fin de generar copias y ambientes de trabajos locales.

Capítulo 2

Objetivos y planificación

2. Objetivos y planificación.

En este apartado se describen los objetivos del proyecto y la planificación realizada para lograr estos objetivos.

2.1 Objetivos

2.1.1 Objetivo general

Desarrollar una aplicación web de gestión de carga horaria utilizando herramientas libres para la carrera de Ingeniería Civil en Bioinformática de la Universidad de Talca.

2.1.2 Objetivo específicos

- Identificar nodos críticos del actual sistema de fichero.
- Contextualizar metodologías de desarrollo ágiles a situación problemática.
- Determinar iteraciones de pequeñas entregas del software web.
- Definir interfaces de usuario de acuerdo para diseño de aplicación web.
- Programar aplicación web requerida para la problemática.
- Realizar validaciones de entregables.

2.2 Plan de trabajo

La plan se estableció en base a un equipo dos personas, rescatándose algunas prácticas de métodos ágiles para el desarrollo de aplicaciones. Las metodologías ágiles son una serie de herramientas basadas en procedimientos, técnicas y documentación que ayudan a desarrolladores a seleccionar recomendaciones más apropiadas para definir y dirigir cada una de las etapas o sub-etapas que conllevan la creación del proyecto de software.

Es común escuchar a practicantes de este tipo de metodologías decir que es algo más que un conjunto de ideas para establecer un paso a paso para desarrollar software, es una filosofía, puesto a las actividades se explican como recomendaciones que pueden ser aplicadas dependiendo del

contexto, cantidad de participantes de un equipo de trabajo, presunción del tamaño del proyecto, organización donde se ejecutará el desarrollo , etc.

Para el desarrollo de este proyecto, se extraerán prácticas de dos tipos de métodos ágiles, eXtreme Programming y Scrum.

2.2.1 eXtreme Programming

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo [6].

XP (imagen 2.2.1) 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. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

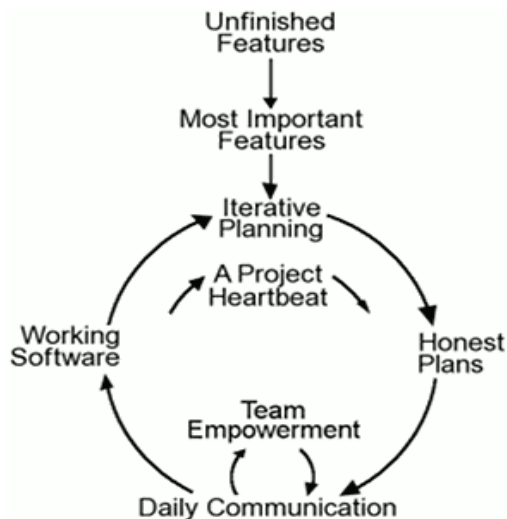


Imagen 2.2.1 – Ciclo XP [<http://www.extremeprogramming.org/>]

2.2.2 Scrum

Propuesto y creado por Ken Schwaber y Jeff Sutherland, Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum (imagen 2.2.2) no es un proceso o una técnica para

construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varios procesos y técnicas. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo de modo que poder mejorar [7].

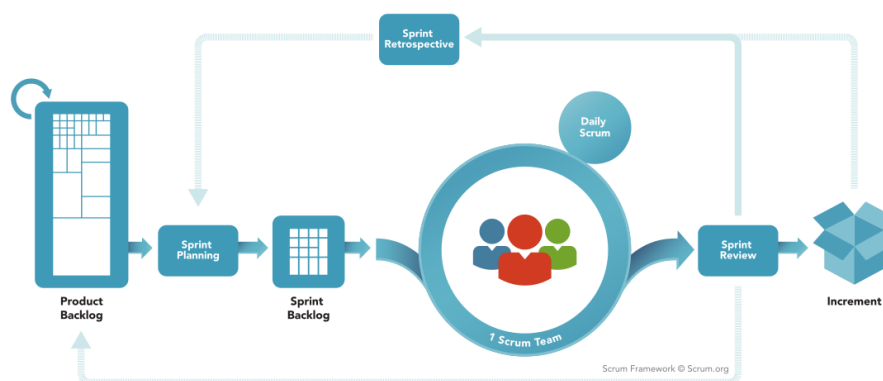


Imagen 2.2.2 – proceso de Scrum [<https://www.scrum.org/Resources/What-is-Scrum>]

La forma de trabajo de Scrum consiste en los Equipos Scrum y sus roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso.

La conclusión a este punto es que una práctica de desarrollo ágil de aplicaciones nos permitirá una solución a los requerimientos para desarrollar software de manera rápida, transparentes para cada uno de los miembros del equipo de trabajo y con gran facilidad de adopción por parte de todos. Otro factor importante para este modo de trabajo se debe a la razón de no entorpecer las labores diarias, especialmente del director de escuela.

2.3 Metodología del Trabajo

Esta se baso en la aceptación por ambas partes (cliente y desarrollador) para construir un trabajo en equipo, siendo esto esencial para aplicar el concepto de metodologías ágiles para el desarrollo de aplicaciones.

Algunas de las prácticas discutidas y aprobadas por los miembros son:

- Juego de la planificación o “*Planning Game*”[9], consiste en un permanente diálogo entre las partes cliente y desarrollador, que fijan objetivos para crear un pequeño entregable.
- Intentar aplicar el concepto de “metáfora” o “épica”, que permita generar historias para que todo el mundo puede conocer el cómo funciona el sistema que ayuden a los usuarios a entender la finalidad del programa.
- Generación de pequeñas entregas (cada diez días), siendo el objetivo en que cada versión debe de ser tan pequeña como fuera posible, conteniendo los requisitos más importantes rescatados de una reunión de planificación.
- Revisión de la iteración una vez concluido el pequeño ciclo definido. Reunión en la que se corrobora el trabajo previsto y que se ha completado y qué parte permanece pendiente para ser abordado en un futuro inmediato. Respecto a lo completado, se realiza una revisión en conjunto a los posibles usuarios que pudiesen estar involucrados, verificando las condiciones de aceptabilidad.
- Retrospectiva de sprint, que será una reunión en la que los miembros del equipo realizan una valoración del trabajo realizado en la última entrega. Buscará mejorar tiempos y formas de arreglos futuros. En otras palabras, es aplicar buenas prácticas en la mejora continua del desarrollo.

La justificación para detallar los puntos anteriores, es por la básica razón de intentar acercar puntos de vista y compromisos para el desarrollo de software.

En esta tratativa se acordaron los siguiente pasos :

- Reunión entre el equipo definido para revisar los planes de desarrollo.
- Asumir los cambios en el desarrollo del software.
- Entregas, revisiones y ajustes de aceptabilidad para el cumplimiento de objetivos.
- Ciclos iterativos hasta que el producto se considera listo para su entrega.

2.4 Carta Gantt

A continuación se representa las expectativas de actividades y tiempos asignados a través de una carta Gantt. Para el diseño de esta carta se han contemplado un número de siete iteraciones, más una entrega final.

Controlado: Fabio Durán Verdugo

Inicio: Octubre 3, 2016

Termino: Enero 8, 2017

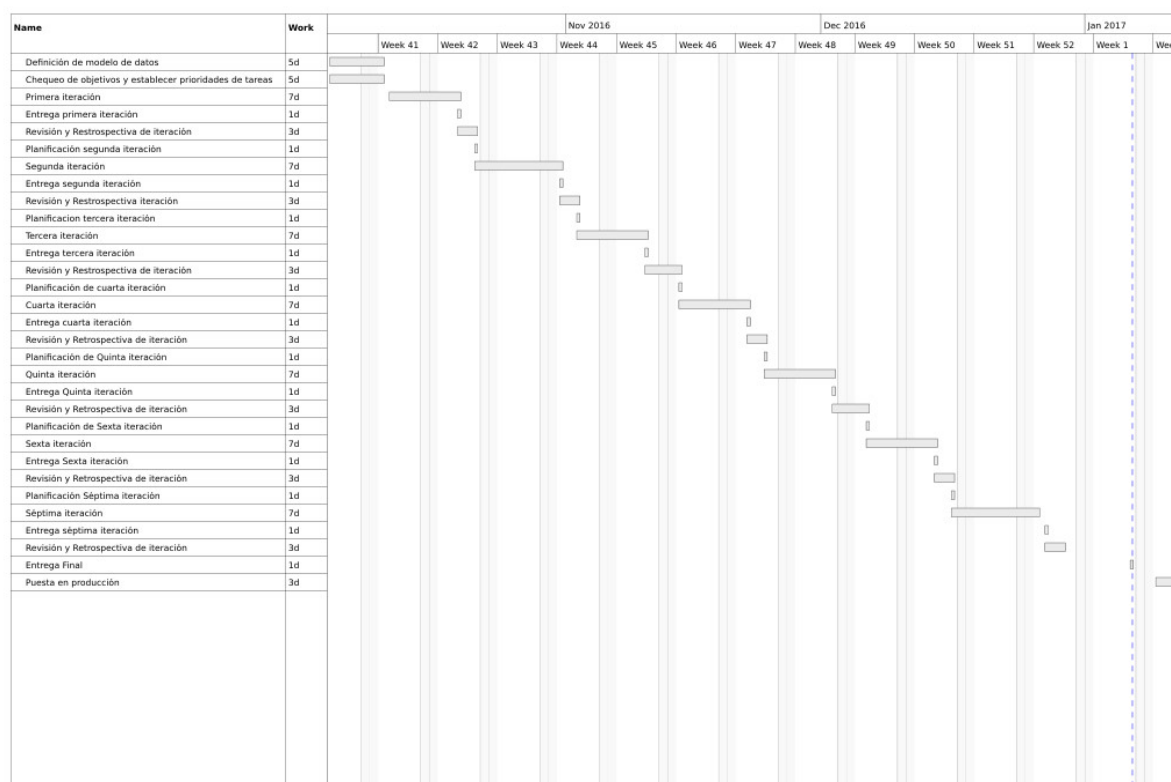


Imagen 2.4 – Carta Gantt del proyecto.

2.5 Equipo de Trabajo

La definición del equipo de trabajo esta dada por la definición y uso de roles fue como lo define Scrum, donde identificaremos a:

- **Product Owner o Cliente**, persona responsable guiar al desarrollador la visión del producto a crear. Su gran responsabilidad es la de definir el conjunto de requerimientos, de priorizarlos y validarlos.

- **Desarrollador**, que es la parte técnica del proceso, recogen las ideas o historias generadas, transformando esto en producto.

También puede tener la participación de generar ideas para el proyecto, así como guiar y apoyar las nociones del cliente afín de sintetizar, comprender ideas a elaborar.

- **Stakeholders o interesados**, serán personas que no forman parte directa del proceso de desarrollo, pero si son personas que incumben en el mismo, pudiendo aportar directa o indirectamente en el desarrollo. Será responsabilidad del Product Owner recoger opiniones y sugerencias y decidir si las aplica a la definición del proyecto, así como también incluir a alguna de estas personas al proceso de revisión del entregables.

- **Usuarios finales**, serán los que accedan a la aplicación una vez liberada.

ROL	Persona	Función o cargo profesional
Product Owner - Cliente	Gabriel Nuñez Vivanco	Director de escuela Ingeniera civil en Bioinformática, Universidad de Talca
Desarrollador	Fabio Durán Verdugo	Estudiante Universitat Oberta Catalunya
Stakeholder - Interesados	-	Asistentes de Escuela
Usuarios Finales	Gabriel Nuñez Vivanco	Director de escuela Ingeniera civil en Bioinformática, Universidad de Talca

Tabla 2.3 – Individualización de roles y personas

Capítulo 3

Requerimientos y solución al problema

3. Requerimientos y solución

En este nivel se determinarían los requisitos y la solución desarrollada, definida en distintos apartados que explicarían los pasos necesarios para la construcción de la aplicación, así como el diseño de las características obtenidas a partir de los relatos.

3.1 Requerimientos

La definición de requerimientos del sistema describen qué es lo que el sistema debe realizar, es decir, un comportamiento esperado del software una vez desarrollado. Para este trabajo la forma provista para obtener los requerimientos a la problemática establecida y el cumplimiento de los objetivos, será mediante la recopilación de relatos del usuario Gabriel Nuñez Vivanco (Product Owner) y en ocasiones intervienen asistentes de escuela (StakeHolder). Estos relatos se adecuarán de modos que definiremos como “Historias de Usuario”.

3.1.1 Concepto de historias de usuarios

Las historias de usuarios son especialmente utilizadas en las metodologías ágiles para la especificación de requisitos, explicándose como una forma de describir lo que el usuario desea ser capaz de hacer, centrándose en el valor que se obtiene por la forma de usar el sistema en lugar de una especificación detallada de lo que el sistema debe hacer. Están concebidos como un medio para fomentar la colaboración entre el desarrollador y el usuario que requiere el desarrollo.

En la práctica se describe de forma limitada, acompañada por una prueba de validación que permita obtener la aprobación del cliente de una forma práctica, visible para la satisfacción para cada requerimiento.

La diferenciación contra los requisitos tradicionales esta dada en como se centran en las operaciones del sistema, que tiende como se mencionó anteriormente a la especificación detallada y compleja del sistema, muchas veces entregando una extenuante documentación incluso exagerada y redundante de los puntos a tratar. Esto también se ve afectado en el modo de comunicación con un cliente sin experiencia, mientras que las historias de los

usuarios se centran en el valor del cliente, utilizando el mismo lenguaje coloquial impuesto por él con una clara y notoria intención de fomentar la comunicación.

Las historias de usuario son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos. Las historias de usuario permiten responder rápidamente a los requisitos cambiantes, por lo que una historia de usuario puede tener muchas mutaciones a lo largo de un desarrollo sin necesariamente verse afectado los tiempos para el mismo [9].

Por último cabe mencionar que uno de los muy buenos beneficios de las historias de usuario es que pueden escribirse en variados niveles de detalle. Se puede escribir una historia que abarque una amplia funcionalidad (relato), siendo esto una “Épica de usuario”.

3.2 Definición de historias de usuarios para el proyecto

La estructura que se ha contemplado para la elaboración de historias de usuarios ha sido la siguiente:

- Un identificador único.
- Un título que identifica el requerimiento a desarrollar.
- La “Épica” es un extracto del relato del usuario que es utilizado para realizar el contexto del requerimiento.
- Las historias de usuario, dividen el comentario de la épica y la plasma como un objetivos por realizar por parte del desarrollador.
- La aceptabilidad, que propone la validación del cumplimiento de los relatos.

3.3 Relatos de historias de usuario

Estas iteraciones será incluidas en la medida de la línea de tiempo en que fueron relatadas, además de estar segmentadas por iteraciones realizadas. Importante de mencionar que cada épica que será expuesta en este punto, va con el intento de expresar fielmente los relatos según fueron enseñados al momento de levantar los requerimientos del sistema.

3.3.1 Primera iteración

Épica: “Cada carrera que está dentro de la universidad puede tener ciertos planes que están de acuerdo a la constante mejora del perfil de estudio, y cada estudiante una vez matriculado se la asocia un plan de estudio tiene su carga académica regularizado por normativa. Estos son los módulos que se debe seguir para obtener su título, una carrera con el paso del tiempo se puede adaptar, esto puede ser que se agregue o cambie una asignatura o sencillamente se cree un nuevo plan que reemplace el actual para los nuevos estudiantes” - Gabriel Nuñez Vivanco

Agregar, modificar, eliminar Carreras	
ID	1
Usuario	Gabriel Nuñez Vivanco
Descripción	Crear formulario que le permita al usuario agregar, modificar, eliminar carreras de una universidad
Aceptabilidad	El usuario logra: <ol style="list-style-type: none"> 1. Realiza una inserción de una carrera. 2. Logra visualizar en una lista de registros de una carrera insertada 3. Logra modificar el nombre de una carrera. 4. Puede eliminar el registro de carrera ingresado.

Agregar, modificar, eliminar planes de estudio y asociarlo a una carrera	
ID	2
Usuario	Gabriel Nuñez Vivanco
Descripción	Crear un formulario que permita al usuario agregar, modificar, eliminar un plan de estudio y asociarlo a una carrera.
Aceptabilidad	El usuario logra: <ol style="list-style-type: none"> 1. Realiza una inserción de un plan de estudio y lo asocia a una carrera. 2. Logra visualizar una lista de registros de una el plan de estudio insertado.

	<ol style="list-style-type: none"> 3. Logra modificar el nombre del plan de estudio que se ha insertado. 4. Logra cambiar el vinculo del plan de estudio a otra carrera ingresada. 5. Puede eliminar el plan de estudio ingresado sin afectar a otros datos relacionado con la carrera.
--	--

Épica: “Poder gestionar cada uno de los módulos, ya sea agregar, modificar, eliminar para distintos planes de estudios, permitiendo también asociarlos según sea la resolución expuesta” - Gabriel Nuñez Vivanco

Agregar, modificar, eliminar módulos y asociarlo a un plan de estudio	
ID	3
Usuario	Gabriel Nuñez Vivanco
Descripción	Se creará un formulario para que el usuario pueda agregar, modificar, eliminar módulos, esto debe incluir un segmento que permita relacionar un módulo a un plan de estudio
Aceptabilidad	El usuario logra: <ol style="list-style-type: none"> 1. Realiza una inserción de un módulo. 2. Logra visualizar en una lista de registros de un módulo insertado 3. Logra modificar el módulo insertado. 4. Eliminar el módulo ingresado sin afectar los datos ingresado para el plan de estudio.

Épica: “En la Universidad existe el concepto llamado departamento al cual se asocian los profesores, esto no tiene mucho que lo relacione con las carreras, puesto a que un departamento puede estar indirectamente relacionado con una carrera o simplemente lo está por un módulo en particular, por ejemplo nosotros tenemos un módulo llamado autogestión del aprendizaje, que pertenece al departamento de vice-rectoría estudiantil, pero nuestra

carrera no tiene más relación con ese departamento que por este ramo. Los profesores se los pedimos a este departamento según nuestra disponibilidad de horario, por lo que no es un problema. A mí me interesa saber de que departamento es qué profesor, para así lograr un sentido y equilibrio al horario” - Gabriel Nuñez Vivanco.

Crear, editar, eliminar departamentos	
ID	4
Usuario	Gabriel Nuñez Vivanco
Descripción	Crear un formulario que le permita al usuario agregar, modificar y eliminar departamentos
Aceptabilidad	El usuario logra: <ol style="list-style-type: none"> 1. Insertar un departamento. 2. Logra visualizar en un lista el departamento agregado. 3. Editar un departamento ya ingresado. 4. Eliminar un departamento insertado.

Crear, editar, eliminar profesores	
ID	5
Usuario	Gabriel Nuñez Vivanco
Descripción	Crear formulario para que al usuario se le permita agregar, modificar y eliminar profesores
Aceptabilidad	El usuario logra: <ol style="list-style-type: none"> 1. Insertar un profesor y relacionarlo a un departamento. 2. Logra visualizar en una lista al profesor ingresado. 3. Editar un profesor ya ingresado. 4. Eliminar un departamento insertado.

3.3.2 Segunda iteración:

Épica: “Necesitamos definir una interfaz de usuario acorde a las políticas de la escuela, así siguiendo los modelos de otros desarrollos realizados por estudiantes, mi propuesta es la siguiente.” - Gabriel Nuñez Vivanco.

Título Aplicación

Modulo

Campo1

Campo2

Campo3

Aceptar Cancelar

Listado de Modulo

Detalle 1	Detalle 2	Detalle 3	Detalle 4
Resultado col 1	Resultado col 2	Resultado col 3	Resultado col 4
Resultado col 1	Resultado col 2	Resultado col 3	Resultado col 4

Imagen 3.3.2 – Propuesta de Gabriel Nuñez Vivanco para formularios para desarrollo del software

Crear interfaz de usuario	
ID	6
Usuario	Gabriel Nuñez Vivanco
Descripción	Se creará una interfaz de usuario según propuesta entregada
Aceptabilidad	El usuario logra: <ol style="list-style-type: none"> 1. Logra visualizar interfaz propuesta notando cada “<i>widgets</i>” que ha sido detallado. 2. Interactúa en todos los formularios desarrollados y comprueba la aplicación de la interfaz.

Épica:“La Universidad divide el horario por bloques de 60 minutos partiendo de las 08:00 AM hasta el último bloque programable que sería las 21:00 horas, esto debemos utilizarlo y seguir la agenda para las asignaturas” - Gabriel Nuñez Vivanco

Crear, editar, eliminar bloques	
ID	7
Usuario	Gabriel Nuñez Vivanco
Descripción	Se deberá crear un formulario que permita al usuario agregar, modificar y eliminar bloques de horario.
Aceptabilidad	El usuario logra: <ol style="list-style-type: none"> 1. Insertar un profesor y relacionarlo a un departamento. 2. Logra visualizar en una lista al profesor ingresado. 3. Editar un profesor ya ingresado. 4. Eliminar un departamento insertado.

3.3.3 Tercera iteración

Épica: “El año calendario de la universidad es a partir de marzo hasta mediados de diciembre y tiene como condición en ser semestral, o sea, esto se divide en dos periodos iguales en las cuales se distribuyen las asignaturas. Por lo tanto cada semestre tiene cinco meses y se agenda un número específico de asignaturas, que estarán de acuerdo a una resolución decretada por cada facultad para su futura calendarización.

Es importante también llevar un histórico de cada calendarización, puesto a que hay profesores que por años realizan el mismo módulo y siempre piden que se les respete el horario del periodo anterior” – Gabriel Nuñez Vivanco

Crear, editar, eliminar años académicos	
ID	8
Usuario	Gabriel Nuñez Vivanco
Descripción	Crear formulario que permita al usuario agregar, modificar y eliminar años académicos
Aceptabilidad	El usuario logra: <ol style="list-style-type: none"> 1. Insertar un año académico 2. Logra visualizar en una lista el año académico

	<p>insertado</p> <ol style="list-style-type: none"> 3. Editar un año académico ya ingresado. 4. Eliminar un año académico insertado.
--	--

Crear, editar, eliminar periodos de un año académico	
ID	9
Usuario	Gabriel Nuñez Vivanco
Descripción	Crear formulario que le permita al usuario agregar, modificar y eliminar periodos de un año académico
Aceptabilidad	<p>El usuario logra:</p> <ol style="list-style-type: none"> 1. Insertar un año periodo para un año académico 2. Logra visualizar en una lista el periodo para el año académico insertado 3. Editar un periodo para un año académico ya ingresado. 4. Eliminar un periodo para un año académico insertado.

Crear, editar, eliminar asignaciones de módulos para un periodo de un año académico contra un profesor	
ID	11
Usuario	Gabriel Nuñez Vivanco
Descripción	Crear formulario que permita al usuario asociar, modificar y eliminar, contra un modulo y profesor para un periodo académico
Aceptabilidad	<p>El usuario logra:</p> <ol style="list-style-type: none"> 1. Agregar una asociación de un modulo, profesor para un periodo de un año académico. 2. Logra visualizar en una lista la asociación creada 3. Editar la asociación de un modulo, profesor para un periodo académico creado.

	4. Eliminar una asociación realizada para un periodo académico sin afectar los módulos, profesor, periodos, ni años académica.
--	--

3.3.4 Cuarta iteración

Épica: “Necesitamos un rediseño de la interfaz, puesto que nos hemos dado cuenta que al momento de insertar todos los módulos de una carrera la lista queda demasiado extensa y el *scroll* de ventana se extiende en demasía, por ejemplo esto afecta a los dispositivos móviles. Debemos pensar en esta tecnología, quizás Bootstrap nos pueda ayudar en crear un sitio *responsive*, Acepto propuestas para esta solicitud, podemos trabajar ambos en diseñar *mockups* para no tener que afectar código mientras decidimos la interfaz definitiva, mi idea es no desviarnos mucho de las aplicaciones que ya poseemos” - Gabriel Nuñez Vivanco

Crear y presentar <i>mockups</i> de interfaz gráfica	
ID	12
Usuario	Gabriel Nuñez Vivanco
Descripción	Crear propuesta de interfaz gráfica para ser presentada y analizada con el director de escuela.
Aceptabilidad	El usuario logra: <ol style="list-style-type: none"> 1. Presenciar y acotar ideas respecto al <i>mockups</i> presentado. 2. Se define y valida diseño de interfaz para aplicar.

Aplicar diseño de interfaz definido en cada módulo diseñado	
ID	13
Usuario	Gabriel Nuñez Vivanco
Descripción	Aplicar el diseño de interfaz de usuario definido para todos los módulos realizados
Aceptabilidad	El usuario logra: <ol style="list-style-type: none"> 1. Apreciar e interactuar con el diseño de interfaz

	definida.
--	-----------

3.3.5 Quinta iteración

Épica “La universidad por normativa decide cada año aplicar horarios protegidos. Esto en realidad son bloques en que no se nos permite programar ni realizar actividades académicas y están destinadas para que los estudiantes organicen actividades propias. Para este año la universidad ha decidido que los bloques 5 y 6 (entre 13:00 y 14:40 horas) y el bloque 10 (20:10 a 21.00 horas) sean protegidos, pero deben ser modificables en el tiempo debido a que no siempre han sido estos bloques los protegidos” - Gabriel Nuñez y Asistente de escuela.

Proponer interfaz para cliente acerca de horarios protegidos	
ID	14
Usuario	Fabio Durán Verdugo
Descripción	Se deberá crear mockups para propuesta de formulario de bloques reservado
Aceptabilidad	El usuario logra: <ol style="list-style-type: none"> 1. Presenciar y acotar ideas respecto al mockups presentado. 2. Definir y validar interfaz para diseñar.

Crear formulario para reserva de bloques protegidos	
ID	15
Usuario	Gabriel Nuñez Vivanco y Asistente de escuela
Descripción	Se deberá crear un formulario que permita al usuario reservar bloques de acuerdo a normativas de la universidad, esto debe ser variable en el tiempo.
Aceptabilidad	El usuario logra: <ol style="list-style-type: none"> 1. Agregar un horario protegido 2. Visualizar los horarios protegidos seleccionados

	3. Modificar horarios protegidos 4. Eliminar los horarios protegidos seleccionados
--	---

3.3.6 Sexta iteración

Épica “Ahora necesitamos crear la interfaz de horario, yo propongo realizar un símil a lo que nosotros utilizamos, o sea tratar de replicar la planilla Excel que poseemos para el caso, lo ideal es poder realizar lo más simple posible la catalogación de un bloque de horario para un día de la semana, creo que hacer “click” para cambiar estados de un bloque puede ser adecuado. Sería bueno que por cada “click” realizado realice el guardado automático en la base de datos.

Recuerda que hay que programar clases, seminarios, ayudantías, talleres y laboratorios, pueden reconocerse con la primera letra de cada palabra, tal cual lo hacemos en la planilla y que son cinco días de la semana todos tienen diez bloques” - Gabriel Nuñez Vivanco

Crear propuesta de mockup para interfaz de usuario de interacción de horario	
ID	16
Usuario	Gabriel Nuñez Vivanco
Descripción	Se deberá crear y presentar un diseño para la interacción del horario a programar, deberá basarse en la planilla de cálculo que se utiliza actualmente.
Aceptabilidad	El usuario logra: <ol style="list-style-type: none"> 1. Presenciar y acotar ideas respecto al “mockup” presentado. 2. Definir y validar interfaz para diseñar.

Crear módulo para gestionar el horario	
ID	17
Usuario	Gabriel Nuñez Vivanco
Descripción	Crear parte del programa que permita la gestión de horarios, esta debe estar basada en la plantilla, así como la

	manipulación. Las acciones estarán basadas en <i>click</i> para cambiar clases (C), seminario (S), ayudantía (A), talleres (T), laboratorio (L) y almacenar en base de datos.
Aceptabilidad	El usuario logra: <ol style="list-style-type: none"> 1. Interacción mediante clicks para el cambio de estado de las actividades (C-S-A-T-L) 2. Salir del formulario y volver a entrar y comprobar que los cambios efectuados dentro del horario fueron realizados

3.3.7 Séptima iteración

Épica: “Definida la interfaz y la interacción, debemos listar los profesores que están seleccionados para un periodo determinado. También sería interesante a través de un “*combobox*” poder cambiar periodos y visualizar horarios anteriores, esto para saber que tan distinto son los horarios entre un año y otro, además de saber de modo rápido el día que hizo clase un profesor.

Hay que validar que los estudiantes de distintos niveles o semestre no se topen entre clases, así como los módulos no permitan programar más clases de las ya definidas en el formulario de módulos. Con esto nos aseguramos que programaremos lo necesario y que no existirán choques de horarios” - Gabriel Nuñez Vivanco

Mostrar profesores y módulos dentro del horario	
ID	18
Usuario	Gabriel Nuñez Vivanco
Descripción	Mostrar a profesores y módulos asociados al periodo correspondiente dentro del formulario de horario.
Aceptabilidad	El usuario logra: <ol style="list-style-type: none"> 1. Visualizar y chequear a los profesores y módulos que corresponden a un periodo seleccionado

Crear “combobox” con periodos para horario	
ID	19
Usuario	Gabriel Nuñez Vivanco
Descripción	Insertar un “ <i>combobox</i> ” en la parte superior de la página que liste todos los periodos y al cambio de opción pueda mostrar el horario que corresponda al periodo seleccionado.
Aceptabilidad	El usuario logra: <ol style="list-style-type: none"> 1. Visualizar y cambiar el periodo de un horario. 2. Ver el horario programado par aun periodo seleccionado.

Crear validación para módulos del mismo nivel	
ID	20
Usuario	Gabriel Nuñez Vivanco
Descripción	Validar que programación de horarios para mismo nivel de módulos no puedan programarse para el mismo día y mismo bloque y arroje mensaje de error
Aceptabilidad	El usuario logra: <ol style="list-style-type: none"> 1. Programar dos módulos del mismo nivel y comprobar que no pueden programarse para el mismo día, mismo bloque. 2. Visualizar mensaje de error

Validar cantidad de horas para actividades académicas	
ID	21
Usuario	Gabriel Nuñez Vivanco
Descripción	Validar dentro de la programación de horarios que la cantidad de horas de clase que pueden programarse no superen las definidas dentro del formulario de módulos.
Aceptabilidad	El usuario logra:

	<ol style="list-style-type: none"> 1. Programar un módulo dentro del horario, cerciorando que la cantidad de horas no sobrepasa la cantidad estipulada, manejado por sistema. 2. Arroja mensaje de error en caso que sobrepase el limite establecido.
--	---

3.4 Interacción para toma de requerimientos

La forma como se ha desarrollado la captura de cada épica, se ha realizado a través de reuniones, como así lo plantea las metodologías ágiles y definido en el punto 2.3.

Estas reuniones denominadas “Planning Game” contemplaban una en cada inicio de iteración para planificar cada una de ellas, en donde se relataba lo que se buscaba obtener como propuesta para el fin del ciclo, y además se especificaba cada una de las pautas de aceptabilidad para validar lo dispuesto.

Se ha definido de tres días para adecuar los conceptos que se han llamado revisión y retrospectiva de iteración, en donde se validaban los objetivos planteados a un comienzo, además de buscar un refinamiento de situaciones particulares que pudiesen ser aplicadas desde la situación en la que se encontraba hacia el futuro del proyecto.

Estas reuniones fueron esenciales en cuanto a agenda en las que participaban el “*Product Owner*” (director de escuela), desarrollador, y en contadas ocasiones con la intervención de “*StakerHolders*” (asistente de escuela).

3.5 Diseño de la solución

En el siguiente se detallarán aspectos del diseño de la construcción de la solución. Incluyen diagramas que ayudarán a comprender mejor la problemática.

3.5.1 Diagrama Entidad Relación ER

A continuación se representa el diagrama ER generado a partir de las historias de usuario descritas anteriormente:

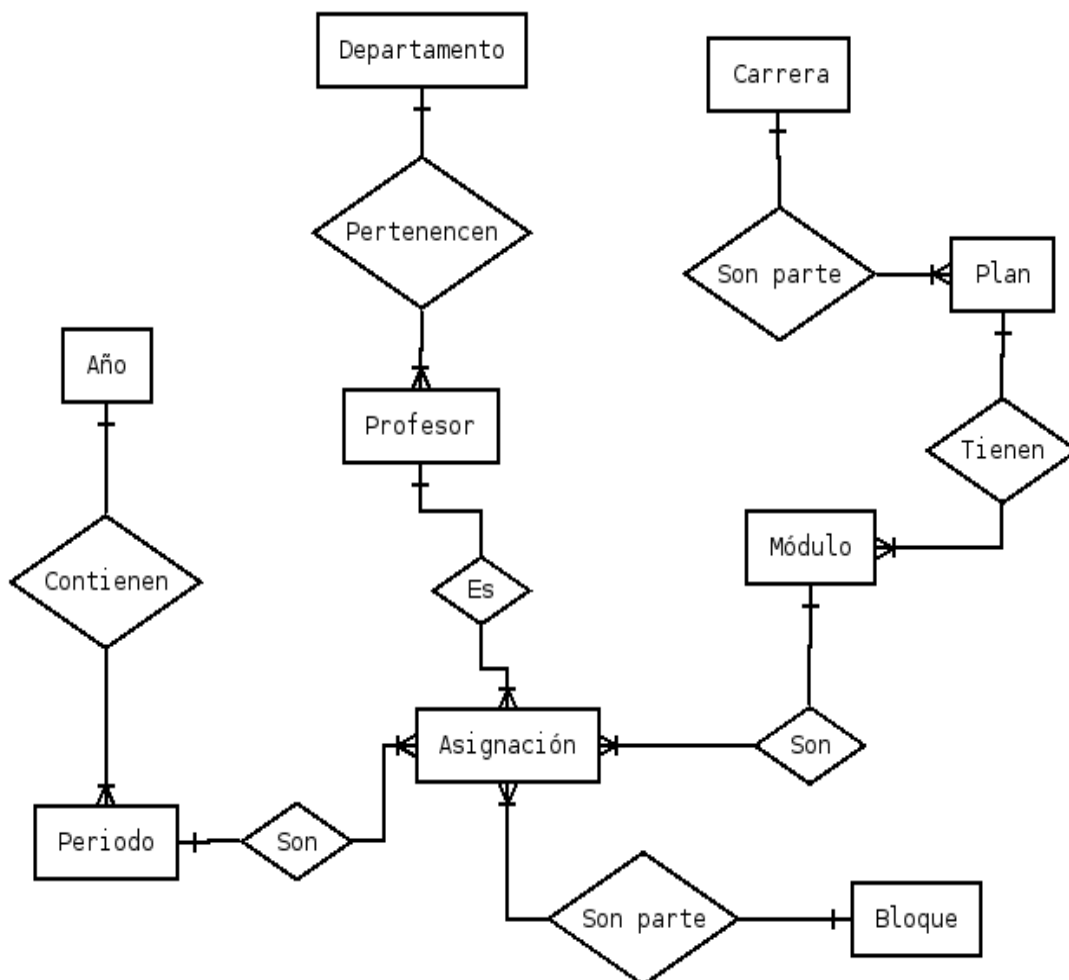


Imagen 3.4.1 – Diagrama entidad relación

3.5.2 Estructura de la aplicación - Diagrama de clases

En este punto se representa el diagrama de clases resultante para la aplicación desarrollada a partir de las historias de usuarios comentadas

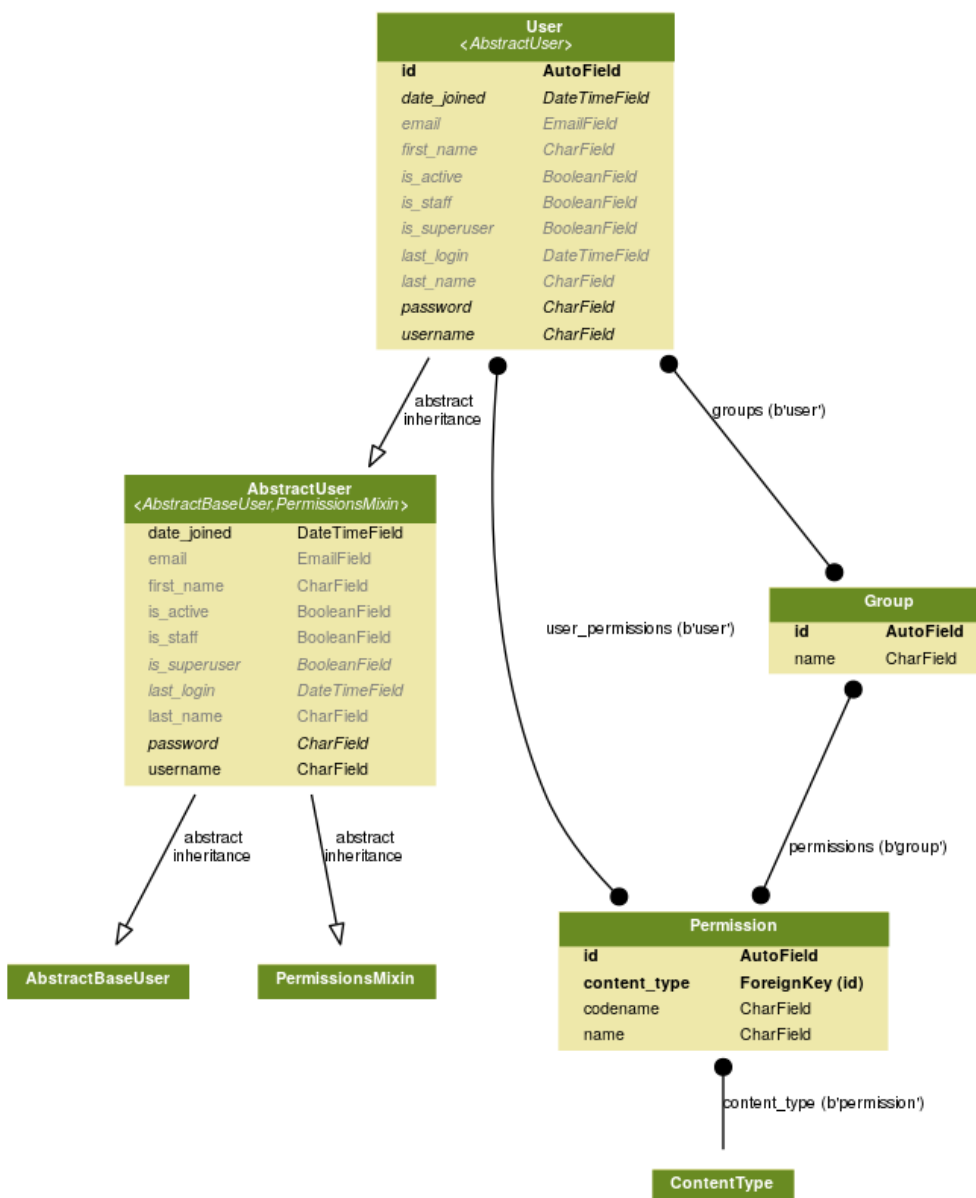


Imagen 3.4.2.1 – Diagrama de clases para aplicación de auth nativa de Django

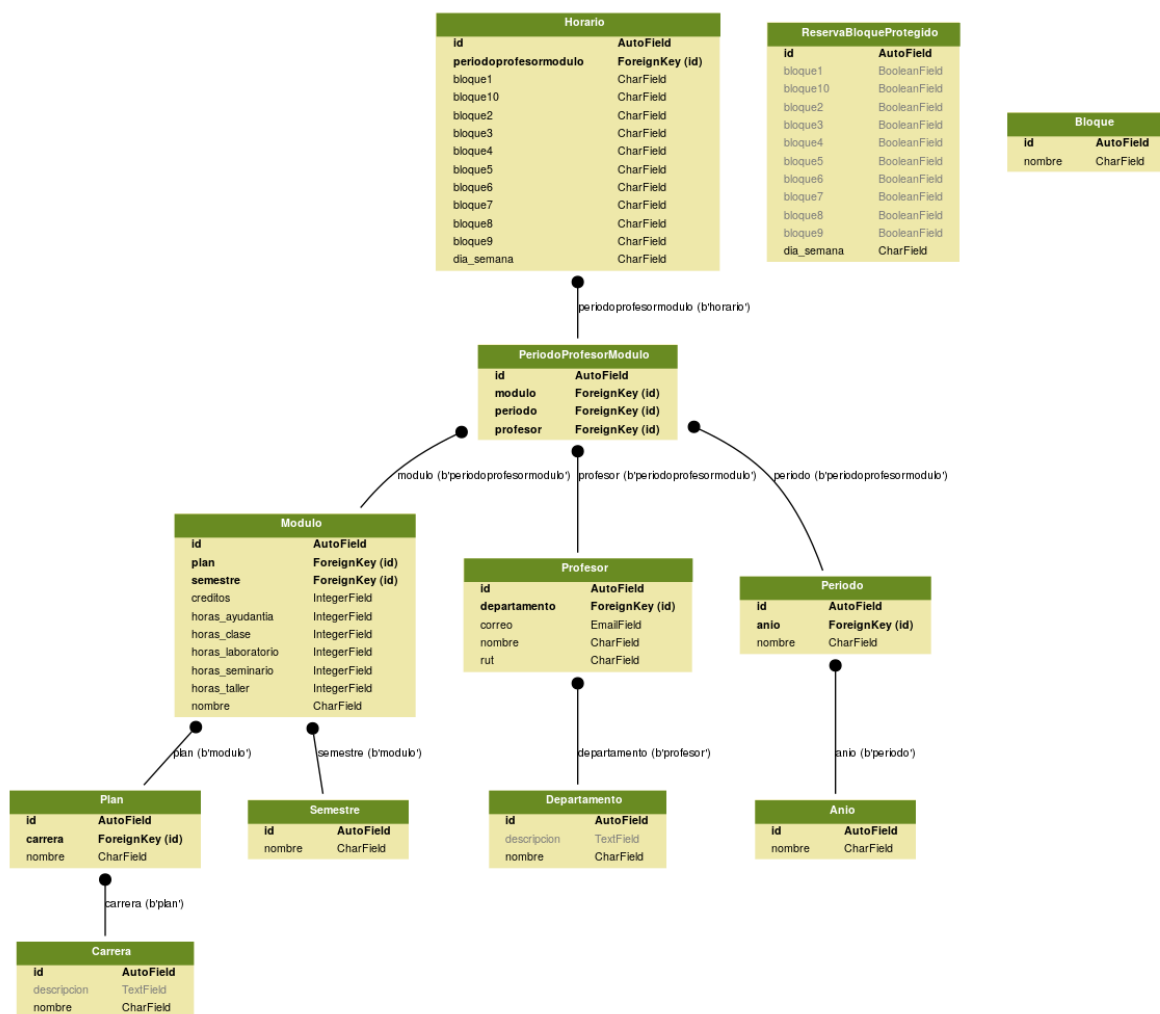


Imagen 3.4.2.2 – Diagrama de clases para aplicación de asignación de horario

3.5.3 Estructura de aplicación

La definición de la aplicación está concentrada en un proyecto denominado “Asignación de horario” y dos aplicaciones del proyecto, una llamada “Horario” y otra llamada “Parámetros” y la representación en árbol sería la siguiente:

```
fabio@rodimusprime:asignacion_horario$ tree -r asignacion_horario/ horario/ parametros/
asignacion_horario/
├── wsgi.py
├── urls.py
├── settings.py
├── __init__.py
horario/
├── views.py
├── valida.py
├── urls.py
├── tests.py
├── models.py
├── __init__.py
├── forms.py
├── errores.py
├── apps.py
├── admin.py
parametros/
├── views.py
├── urls.py
├── tests.py
├── sectioned_form.py
├── models.py
├── __init__.py
├── forms.py
├── apps.py
├── admin.py

0 directories, 23 files
```

Imagen 3.3.4 – Árbol de directorio del proyecto.

3.5.4 Definición de modelo en Django

El archivo a representar es `models.py` de cada aplicación, en donde un fragmento de la estructura de codificación es la siguiente:

```
from django.db import models

class Plan(models.Model):
    nombre = models.CharField(max_length=10)
    carrera = models.ForeignKey(Carrera, null=True)

    class Meta:
        ordering = ["-id"]

    def __str__(self):
        return "%s" % (self.nombre)

class Semestre(models.Model):
    nombre = models.CharField(max_length=20)
```

```

def __str__(self):
    return self.nombre

class Modulo(models.Model):
    nombre = models.CharField(u'Modulo', max_length=100)
    semestre = models.ForeignKey(Semestre, default=0)
    plan = models.ForeignKey(Plan)
    creditos = models.IntegerField(u'Créditos', default=0)
    horas_clase = models.IntegerField(u'Horas de clase', default=0)
    horas_seminario = models.IntegerField(u'Horas de seminario', default=0)
    horas_laboratorio = models.IntegerField(u'horas de laboratorio', default=0)
    horas_taller = models.IntegerField(u'Horas de Taller', default=0)
    horas_ayudantia = models.IntegerField(u'Horas de Ayudantía', default=0)

def __str__(self):
    return ("%s del %s"%(self.nombre, self.plan.nombre))

```

3.5.5 Definición de formularios

Cada uno de los formularios está definido en el fichero forms.py y se muestra como ejemplo la forma en como se ha codificado:

```

from django import forms
from .sectioned_form import SectionedForm
from .models import (
    Carrera,
    Departamento,
    Plan,
    Modulo,
    Anio,
    Periodo,
    Bloque,
    Profesor,
    Semestre,
)

class ModuloForm(forms.ModelForm):

    class Meta:
        model = Modulo

```

```

fields = ["nombre",
          "semestre",
          "plan",
          "horas_clase",
          "horas_seminario",
          "horas_laboratorio",
          "horas_taller",
          "horas_ayudantia",
          ]

```

3.5.6 Definición de Vistas en Django

Las vistas han sido definidas bajo el concepto de Class-based views de Django y el ejemplo de como se ha definido la vista para el formulario de módulo ha sido el siguiente:

```

class StaffRequiredMixin(object):
    @method_decorator(staff_member_required)
    def dispatch(self, request, *args, **kwargs):
        return super(StaffRequiredMixin, self).dispatch(request, *args, **kwargs)

class ViewCreateView(SuccessMessageMixin, CreateView):
    form_class = None
    template_name = None
    success_message = None
    titulo = None

    def get_context_data(self, **kwargs):
        context = super(ViewCreateView, self).get_context_data(**kwargs)
        context['titulo'] = self.titulo
        return context

    def get_success_message(self, cleaned_data):
        #cleaned_data is the cleaned data from the form which is used for string
        #formatting
        return self.success_message % dict(cleaned_data,
                                           nombre=self.object.nombre)

class ModuloCreateView(StaffRequiredMixin, ViewCreateView):
    form_class = ModuloForm
    template_name = "parametros/form.html"

```

```

titulo = "Agrega Modulo"
success_message = "El Modulo %(nombre)s ha sido creado"
success_url = "/modulo/"

```

3.5.7 Definición de los Templates

```

{% block content %}

<div class="row">

  <div class="col-lg-12">
    <div class="header">
      <h1>{{ titulo }}</h1>
      <div class="col-lg-10"></div>
      <div class="col-lg-2">
        <button class="btn btn-success add">Añadir {{ titulo }}</button>
      <br></br>
    </div>
  </div>
</div>

<div class="col-lg-12">
  <div class="panel panel-default">
    <div class="panel-heading">
      {{ titulo }}
    </div>
    <!-- /.panel-heading -->
    <div class="panel-body">
      <table width="100%" class="table table-striped table-bordered table-
hover" id="dataTables-example">
        <thead class="thead-default">
          <tr>
            <th>Modulo</th>
            <th>Plan</th>
            <th>Semestre</th>
            <th>Carrera</th>
            <th>Clases</th>
            <th>Seminario</th>
            <th>Laboratorio</th>
            <th>Taller</th>

```

```

        <th>Ayudantía</th>
        <th></th>
        <th></th>
    </tr>
</thead>
<tbody>
{% for obj in object_list %}
    <tr>
        <td>{{ obj.nombre }}</td>
        <td>{{ obj.plan }}</td>
        <td>{{ obj.semestre.nombre }}</td>
        <td>{{ obj.plan.carrera }}</td>
        <td>{{ obj.horas_clase }}</td>
        <td>{{ obj.horas_seminario }}</td>
        <td>{{ obj.horas_laboratorio }}</td>
        <td>{{ obj.horas_taller }}</td>
        <td>{{ obj.horas_ayudantia }}</td>
        <td><p data-placement="top" data-toggle="tooltip"
title="Edit"><button class="btn btn-primary btn-xs edit" data-title="Edit" data-
toggle="modal" data-target="#edit" value="{{ obj.id }}" ><span class="glyphicon
glyphicon-pencil"></span></button></p></td>
        <td><p data-placement="top" data-toggle="tooltip"
title="Delete"><button class="btn btn-danger btn-xs delete" data-title="Delete" data-
toggle="modal" data-target="#delete" value="{{ obj.id }}"><span class="glyphicon
glyphicon-trash"></span></button></p></td>
    </tr>
{% endfor %}
</tbody>
</table>
<!-- /.table-responsive -->
</div>
<!-- /.panel-body -->
</div>
<!-- /.panel -->
</div>
<!-- /.col-lg-12 -->
</div>
<!-- /.row -->
{% endblock %}

```

3.6 Evolución de la interfaz de usuario

A continuación se representa de forma secuencial la evolución y los cambios de la interfaz de usuario propuesta por el “*Product Owner*” siguiendo los paradigmas de otras aplicaciones creadas en la escuela de Ingeniería Civil en Bioinformática para la búsqueda de uniformidad.

3.6.1 Aplicación para parámetros

- Primer modelo propuesto para formulario de aplicación de parámetros:

El formulario muestra un título "Titulo Aplicación" en un recuadro superior. A la izquierda hay un menú con siete ítems etiquetados como "Ítem Menú". A la derecha, bajo el título "Modulo", hay tres campos de entrada etiquetados como "Campo1", "Campo2" y "Campo3", seguidos de botones "Aceptar" y "Cancelar". Debajo de esto, se encuentra un "Listado de Modulo" que es una tabla con cuatro columnas: "Detalle 1", "Detalle 2", "Detalle 3" y "Detalle 4". La tabla contiene dos filas de datos, cada una con "Resultado col 1", "Resultado col 2", "Resultado col 3" y "Resultado col 4".

imagen 3.6.1.1 – Propuesta del primer modelo de interfaz para formularios de parámetros

Se puede definir como un formulario único que permite la interacción del usuario y a la vez ver resultados en la misma pantalla. El problema suscitado resalta cuando existe un cantidad mayor de 5 de ítem provoca tener que interactuar con el “*scroll*” de ventana o al tener que editar se hacía necesario volver arriba al formulario. Esto fue declarado en la historia de usuario con identificador 12 (cuarta iteración).

- Segundo modelo propuesto y final para formularios de aplicación de parámetros:

Título de la aplicación

Menú

[Ítem Menú](#)

[Ítem Menú](#)

[Ítem Menú](#)

[Ítem Menú](#)

[Ítem Menú](#)

[Ítem Menú](#)

[Ítem Menú](#)

Nombre del Módulo

Mensaje Post acción
Nuevo Item

<input type="checkbox"/>	Column 1	Column 2	Column 3	Column 4	Column 5
<input checked="" type="checkbox"/>	Cell Content 1	Cell Content 2	Cell Content 3	Cell Content 4	Edit Delete
<input type="checkbox"/>	Cell Content 1	Cell Content 2	Cell Content 3	Cell Content 4	Edit Delete
<input type="checkbox"/>	Cell Content 1	Cell Content 2	Cell Content 3	Cell Content 4	Edit Delete
<input type="checkbox"/>	Cell Content 1	Cell Content 2	Cell Content 3	Cell Content 4	Edit Delete
<input type="checkbox"/>	Cell Content 1	Cell Content 2	Cell Content 3	Cell Content 4	Edit Delete
<input type="checkbox"/>	Cell Content 1	Cell Content 2	Cell Content 3	Cell Content 4	Edit Delete

Imagen 3.6.1.2 - Segunda propuesta del modelo de interfaz para formularios de parámetros, listados

Título de la aplicación

Menú

[Ítem Menú](#)

[Ítem Menú](#)

[Ítem Menú](#)

[Ítem Menú](#)

[Ítem Menú](#)

[Ítem Menú](#)

[Ítem Menú](#)

Label Accion (Añadir - Editar) Modulo

Campo 1

Campo 2 ▼

Campo 3

Campo 4

Text box

Cancelar
Aceptar

Imagen 3.6.1.3 - Segunda propuesta del modelo de interfaz para formularios de parámetros, añadir, editar

En este modelo de interfaz propone dos ventanas que reducen el movimiento de “*scroll*” por parte del usuario, manteniendo todos los datos requeridos a la vista. La cantidad de ítem son filtrados por un complemento de Bootstrap que permite listar un número de registros limitados y manejado por usuarios.

3.6.2 La aplicación Horario

- Primer modelo propuesto para el horario es el siguiente:

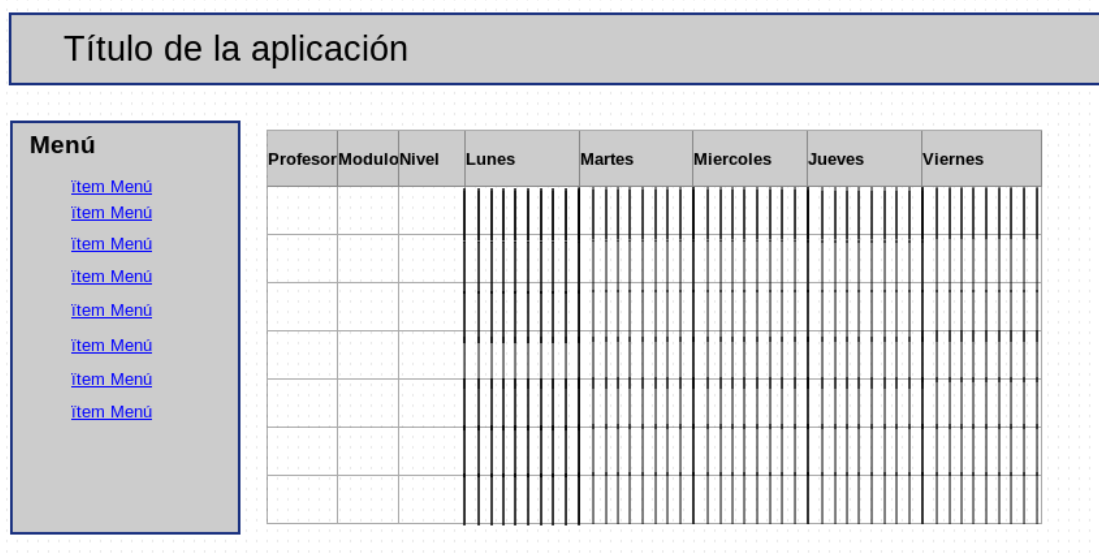


Imagen 3.6.2.1 – Primera propuesta para interfaz de creación de horario

La idea de este modelo es replicar la planilla de cálculo expuesta en el punto 1.2 (situación actual), pero la realidad del “*mockup*” no era posible replicarla en el template HTML de la aplicación y la expansión de la ventana hacia la derecha era extremadamente alargada.

- Segundo modelo y propuesta final



Imagen 3.6.2.2 – Propuesta final para interfaz de creación de horario

Esta última propuesta contempla la generación de los bloques y una separación por día de la semana. A pesar que la extensión puede ser prolongada hacia abajo, se hace más natural el trabajo por parte de los usuarios, por lo que se determinó implementar esta última opción.

3.7 Extensiones extras de Django utilizadas

Para completar la propuesta se ha hecho necesario reutilizar algunas herramientas para facilitar el desarrollo de la aplicación. Estas extensiones son:

- **Django-extensions:** una colección de extensiones personalizadas para Django, que incluyen comandos de administración, campos de base de datos adicionales, extensiones de administración entre otras.

La licencia por la que se rige es MIT.

- **Django-bootstrap3:** extensión para que permite integrar las características y habilidades de bootstrap3 a Django. Licenciado bajo MIT.

3.8 Extensión extras de Bootstrap3

Se ha utilizado una extensión llamada DataTable para facilitar el desarrollo y la interacción con tablas, para este caso en cada módulo que se requiera mostrar los ítem insertados en la base de datos. La licencia que utiliza esta extensión es MIT.

3.9 Repositorio

El software se ha desarrollado bajo el control de Git y alojado en GitHub, la url para acceder al él es: https://github.com/hbkfabio/asignacion_horario

3.10 Pendientes por resolver

Algunos aspectos pendientes por resolver y acordados con el product Owner son los siguientes:

- Criterios de usabilidad e interacción con la interfaz. Esto es una problemática que como escuela no se ha logrado consensuar y este proyecto no está ajeno a ello. Mientras se debe seguir las métricas definidas.
- Se despliega del punto anterior la búsqueda de una forma de mostrar más información respecto a los módulos y profesores para la planificación del horario, sin tener que saturar la pantalla con elementos innecesarios.
- Validaciones de distintos niveles para planes de estudios en caso de tener a dos profesores dictando en mismo módulo pero en grupos distintos. Para este caso el horario debiese permitir agendar.
- Generación de reportes varios.
- Cuadro de búsqueda interactiva de menú.

Capítulo 4

Resultados

4. Resultados

En este ítem se demostraran los resultados alcanzados específicamente por desarrollo de la aplicación. Se detallará a través de imágenes que presentarán los alcances logrados.

4.1 Visualización de la aplicación

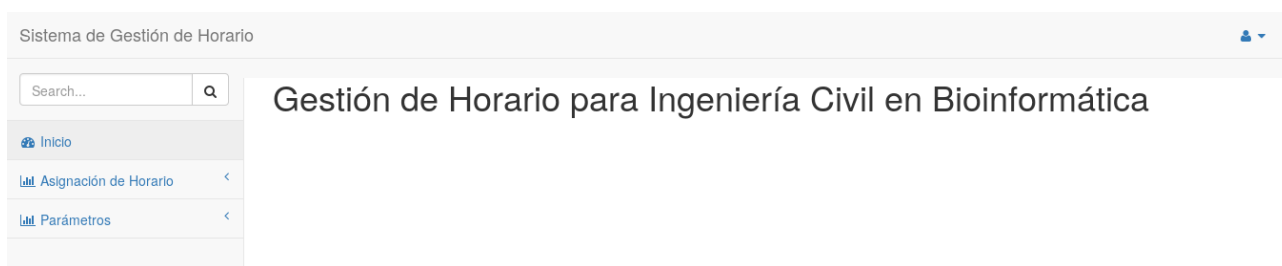


Imagen 4.1(1) Portada de la aplicación

En estas imágenes se puede apreciar la portada de la aplicación, pudiéndose resumir en un menú lateral con las opciones y un cuadro de búsqueda de entrada de menú.

Todas las entradas para la aplicación de parámetros tendrán las mismas formas y entradas como se muestra en la imagen de a continuación:

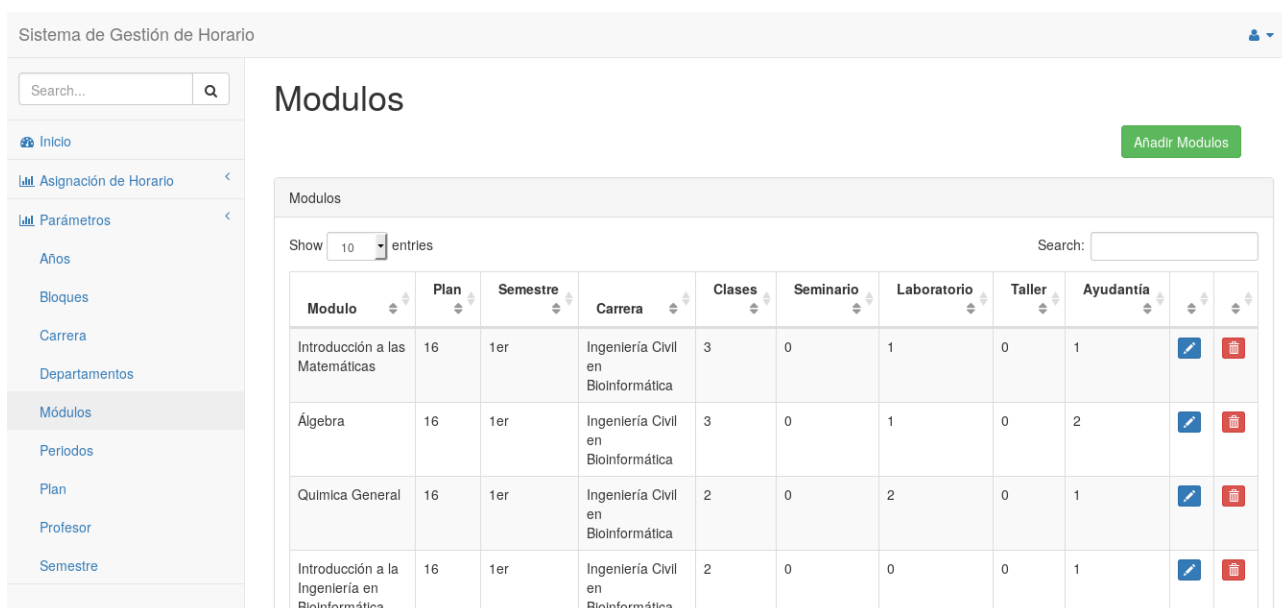
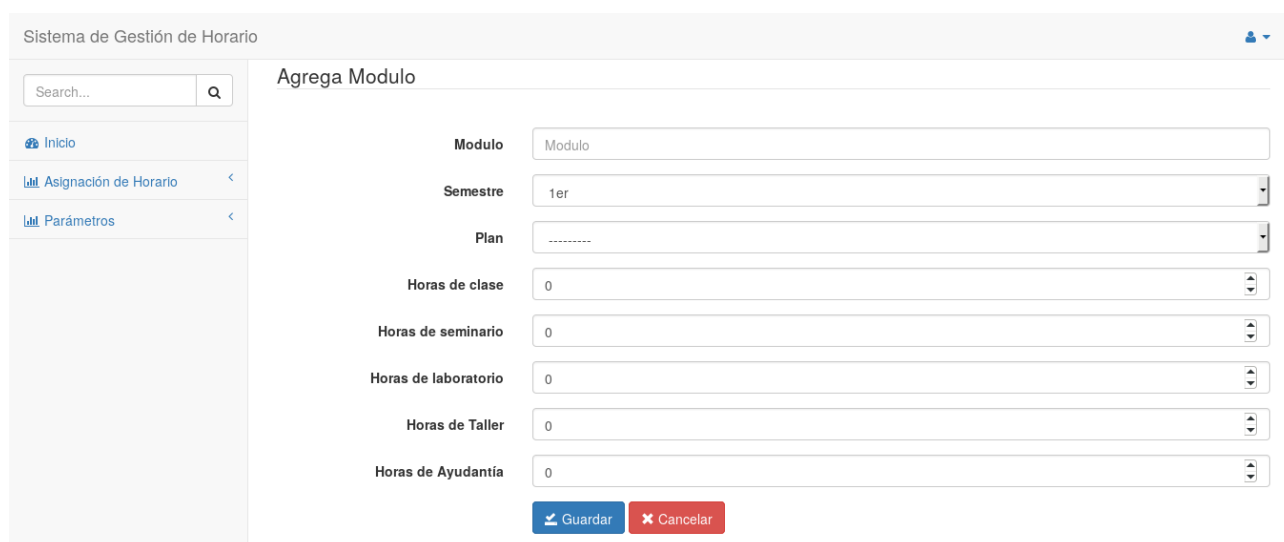


Imagen 4.1(2) – Listado de ítem para aplicación de parámetros

En esta se puede apreciar un listado con todos los ítem ingresados y cada uno con un botón de color azul para editar y de color rojo para eliminar. Dentro de esta tabla se puede ver la extensión de “*DataTable*” que incluye un filtro dinámico de búsqueda que permite mostrar resultados a medida se va tecleando alguna palabra.

Otra característica importante que se puede limitar, es la cantidad de resultados de forma dinámica, logrando que esto no acapare toda la pantalla con resultados, y la vista del usuario sea cómoda para apreciar los resultados sin tener que manipular el “*scroll*”.

Si se presiona el botón de color verde ubicado en la esquina superior derecha de la pantalla se podrá acceder a la siguiente pantalla:



Sistema de Gestión de Horario

Search...

Inicio

Asignación de Horario

Parámetros

Agrega Modulo

Modulo

Semestre

Plan

Horas de clase

Horas de seminario

Horas de laboratorio

Horas de Taller

Horas de Ayudantía

Imagen 4.1(3) – Agregando un ítem para la aplicación de parámetros

Acá lo que podremos realizar es el completado de todos los atributos que el requerimiento ha entregado y guardar dentro de nuestra base de datos.

A esta opción también podemos acceder desde el botón editar de la primera pantalla, la diferencia radica en que la pantalla ahora incluirá los datos que necesitamos modificar.

Sistema de Gestión de Horario

Search...

Inicio

Asignación de Horario

Parámetros

None

Modulo: Quimica General

Semestre: 1er

Plan: 16

Horas de clase: 2

Horas de seminario: 0

Horas de laboratorio: 2

Horas de Taller: 0

Horas de Ayudantía: 1

Imagen 4.1(4)– Actualizar datos para aplicación de parámetros

Dentro de la otra aplicación de esta aplicación web llamada horario resolver que la entrada de horario se definió una interfaz con el cliente, siendo el resultado el siguiente:

Sistema de Gestión de Horario

Search...

Inicio

Asignación de Horario

Asignar Profesor Módulo

Horario

Bloques Protegidos

Parámetros

Horario

2017 Primer Semestre

C=Clases, A=Ayudantía, S=Seminario, L=Laboratorio, T=Taller

Lunes

Nombre	Modulo	Plan	Semestre	1	2	3	4	5	6	7	8	9	10

C=Clases, A=Ayudantía, S=Seminario, L=Laboratorio, T=Taller

Martes

Nombre	Modulo	Plan	Semestre	1	2	3	4	5	6	7	8	9	10

C=Clases, A=Ayudantía, S=Seminario, L=Laboratorio, T=Taller

Miercoles

Nombre	Modulo	Plan	Semestre	1	2	3	4	5	6	7	8	9	10

Imagen 4.1(5) – Formulario de horario vacío

Para ser llenado esto se debe dar de alta a los profesores y módulo para un periodo determinado y definido por el usuario. Esta opción es “Asignar Profesor Módulo”



Imagen 4.1(6) – Listado Asignar Profesor Módulo

La interfaz sigue la lógica de parámetros.

Pero al momento de añadir una acción se debe seleccionar un profesor contra un módulo para un periodo determinado.

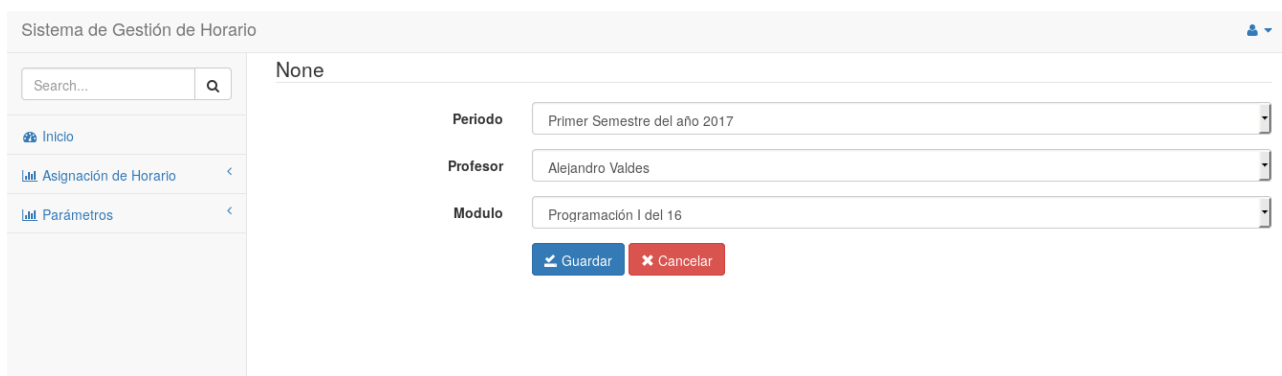


Imagen 4.1(7) – Añadir o editar Asignar Profesor Módulo

Otra de las entradas es la reserva de bloques reservados (historia de usuario 14 y 15)

Sistema de Gestión de Horario

Search...

[Inicio](#)

[Asignación de Horario](#)

- Asignar Profesor Módulo
- Horario
- Bloques Protegidos

[Parámetros](#)

Agrega Reserva de Bloque Protegido

- Bloque1
- Bloque2
- Bloque3
- Bloque4
- Bloque5
- Bloque6
- Bloque7
- Bloque8
- Bloque9
- Bloque10

Imagen 4.1(8) – Formulario de Bloques protegidos

Sistema de Gestión de Horario

Search...

[Inicio](#)

[Asignación de Horario](#)

- Asignar Profesor Módulo
- Horario
- Bloques Protegidos**

[Parámetros](#)

Reserva Bloques protegidos Módulo

Reserva Bloques protegidos Módulo

Show entries Search:

Bloque 1	Bloque 2	Bloque 3	Bloque 4	Bloque 5	Bloque 6	Bloque 7	Bloque 8	Bloque 9	Bloque 10	
				X	X				X	<input type="button" value="✎"/>

Showing 1 to 1 of 1 entries

Previous **1** Next

Imagen 4.1(9) – Formulario de Bloques protegidos resultado post-guardado

La funcionalidad de “reservar bloques” será dentro de un periodo determinado y modificable en el tiempo. Una vez guardado el resultado se verá reflejado en el horario de la aplicación.

Sistema de Gestión de Horario

Search...

[Inicio](#)

[Asignación de Horario](#)

[Parámetros](#)

Horario

2017 Primer Semestre

C=Clases, A=Ayudantía, S=Seminario, L=Laboratorio, T=Taller

Lunes

Nombre	Modulo	Plan	Semestre	1	2	3	4	5	6	7	8	9	10
Alejandro Valdes	Programación I	16	2do					X	X				X

C=Clases, A=Ayudantía, S=Seminario, L=Laboratorio, T=Taller

Martes

Nombre	Modulo	Plan	Semestre	1	2	3	4	5	6	7	8	9	10
Alejandro Valdes	Programación I	16	2do					X	X				X

C=Clases, A=Ayudantía, S=Seminario, L=Laboratorio, T=Taller

Imagen 4.1(10) – Horario

El resultado de completar los ítem anteriores es la muestra de la imagen anterior, un Horario con profesores, módulos y dispuestos a ser programados. Cabe notar las “X” en los bloques 5, 6, y 10 que son los reservados, visto en el apartado de reserva de bloques protegidos.

La interacción del horario es mediante “click” dentro de cada cuadrado de bloque disponible. Cada interacción arrojará una letra ya definida para la acción (C, A, S, L, T). Fijarse en los bloques 1 del día lunes y 3 del día martes.

Sistema de Gestión de Horario

Search...

[Inicio](#)

[Asignación de Horario](#)

[Parámetros](#)

Horario

2017 Primer Semestre

C=Clases, A=Ayudantía, S=Seminario, L=Laboratorio, T=Taller

Lunes

Nombre	Modulo	Plan	Semestre	1	2	3	4	5	6	7	8	9	10
Alejandro Valdes	Programación I	16	2do	C				X	X				X

C=Clases, A=Ayudantía, S=Seminario, L=Laboratorio, T=Taller

Martes

Nombre	Modulo	Plan	Semestre	1	2	3	4	5	6	7	8	9	10
Alejandro Valdes	Programación I	16	2do			L		X	X				X

C=Clases, A=Ayudantía, S=Seminario, L=Laboratorio, T=Taller

Imagen 4.1(11) – Interacción con Horario

Existe una validación para saber la cantidad de actividades programables para un módulo determinado, cuyos valores están regidos por el formulario de parámetros – módulos, a fin de poder programar solo que está disponible dentro de estas opciones. En caso de sobrepasar los límites lanzará un mensaje de advertencia y no almacenará la opción en la base de datos (historia de usuario 21)

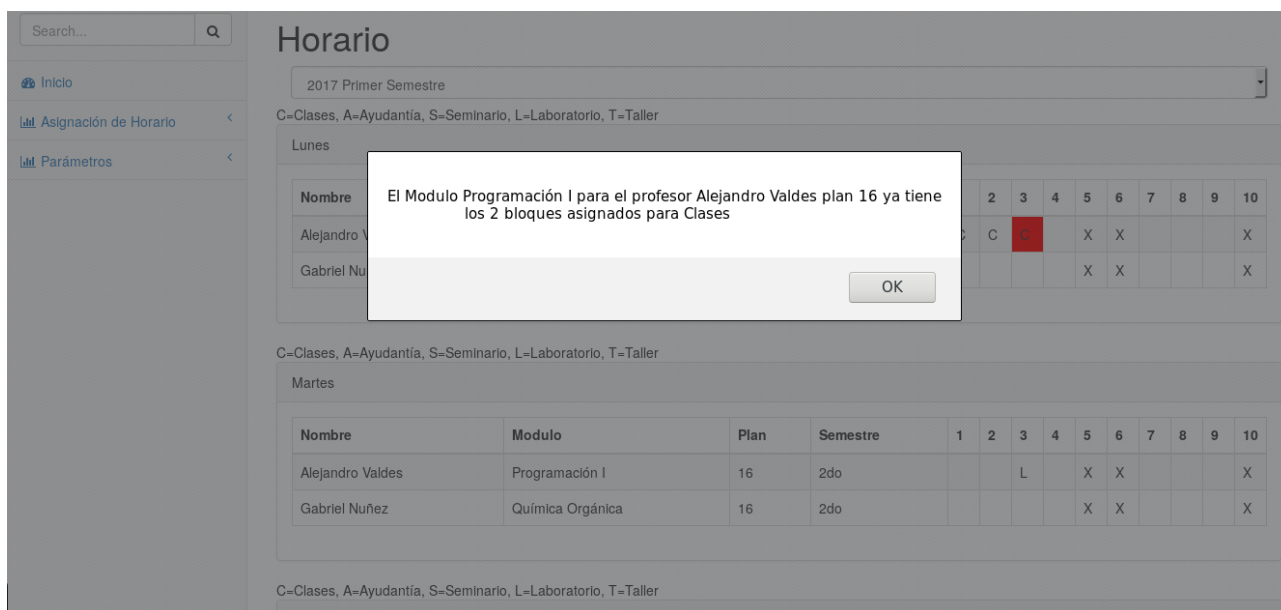


Imagen 4.1(12) – Interacción con Horario con mensaje de error por sobrepaso limite de opciones

Lo mismo sucederá para la validación de la historia de usuario 20

Search...

Horario

2017 Primer Semestre

C=Clases, A=Ayudantía, S=Seminario, L=Laboratorio, T=Taller

Lunes

Nombre	1	2	3	4	5	6	7	8	9	10
Alejandro Valdes	C	C			X	X				X
Gabriel Nuñez					X	X				X

Existe un choque de horario para el bloque seleccionado

OK

C=Clases, A=Ayudantía, S=Seminario, L=Laboratorio, T=Taller

Martes

Nombre	Modulo	Plan	Semestre	1	2	3	4	5	6	7	8	9	10
Alejandro Valdes	Programación I	16	2do			L		X	X				X
Gabriel Nuñez	Química Orgánica	16	2do					X	X				X

C=Clases, A=Ayudantía, S=Seminario, L=Laboratorio, T=Taller

Imagen 4.1(13) – Interacción con Horario con mensaje de error por choque de horario

Capítulo 5

Conclusiones

5 Conclusiones

5.1 Cumplimiento del objetivo

Se ha alcanzado el objetivo general previsto, más como una aplicación web dinámica perfectible, queda trabajo para alcanzar la plenitud basada en la experiencia de usuario que requiere experimentar mejores prácticas para explotar de mejor manera las interfaces de usuarios en pro de la búsqueda de mejorar la experiencia de usuario.

En lo referente a los objetivos específicos, estos tuvieron una concreción positiva ya que la planificación, diseño y ejecución permitieron concretar el desarrollo de la aplicación web.

5.2 Conclusiones personales

El desarrollo de este proyecto ha sido satisfactorio, porque permitió aportar con conocimientos basados en la computación e informática en la construcción de una aplicación que responde a un vacío existente en una institución.

A pesar de que se pueda considerar una aplicación básica, puede permitir crear las primeras bases para que se puedan desarrollar futuros programas en mismo ámbito. Esto incluye la posibilidad de poder crear comunidad alrededor del software donde se pueda tomar un mayor número de requisitos, experiencias y conocimientos de ambientes web, para construir un producto integral y oficial para la universidad u otra institución educativa.

5.3 Conocimientos aplicados y aprendizajes

El proyecto se ha realizado dentro del área de desarrollo de software, por lo que los conocimientos relativos a las asignaturas y aprovechados de manera personal serían: Base de datos, Desarrollo de aplicaciones, Desarrollo de aplicaciones Web.

Es posible incluir en este apartado los conocimientos adquiridos en aspecto económicos del software libre, puesto a que lo más importante que se ha resaltado en conocimientos, es la búsqueda incesante por formar comunidad y tratar de llevar un producto de software al éxito.

5.4 El futuro

Es de esperar:

- De crear comunidad alrededor del software y poder generar más requisitos y opciones de desarrollo.
- Mejorar experiencias de usuarios.
- Generación de reportes para variadas situaciones.
- Pulir errores resultantes y no contemplados o encontrados en la actualidad.
- Lograr ser un software oficial de la universidad.

Capítulo 6

Referencias

6. Referencias

- [1] Descripción de Ingeniería Civil en Bioinformática - <http://icb.utalca.cl/html/descripcion.html> – Consultado el 04/01/2017
- [2] Universidad de Talca - https://es.wikipedia.org/wiki/Universidad_de_Talca – Consultado el 04/01/2017
- [3] Django project - <https://www.djangoproject.com/> - Consultado el 04/01/2017
- [4] PostgreSQL - <https://www.djangoproject.com/> - Consultado el 04/01/2017
- [5] Git - <https://git-scm.com/> - Consultado el 05/01/2017
- [6] K. Beck. Extreme Programming Explained: Embrace Change. Addison-Wesley, Pearson Education, 2000. ISBN 201-61641-6]
- [7] La Guía de Scrum - Ken Schwaber y Jeff Sutherland - pagina 3 - <http://www.scrumguides.org/>
- [8] eXtreme Programming Explained: Embrace Change - Kent Beck – 2002. ISBN 201-61641-6
- [9] eXtreme Programming Explained: Embrace Change pag 86 - Kent Beck – 2002. ISBN 201-61641-6
<https://books.google.es/booksid=G8EL4H4vf7UC&printsec=frontcover&hl=es#v=onepage&q=User&f=false>

Capítulo 7

Anexos

7.1 – Licencia del Documento publicado

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not

explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years

before the Document itself, or if the original publisher of the version it refers to gives permission.

- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty

Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.