

CIS1830CP04

Aplicación web para la planeación de horarios académicos en la PUJ - JPlanner

Juan Sebastián Gómez

Nicolas Restrepo

Álvaro David Pongutá

Juan Felipe Hernández

Jesús Gilberto Pineda

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

SYSTEMS ENGINEERING PROGRAM

BOGOTÁ, D.C.

2019

CIS1830CP04

Aplicación web para la planeación de horarios académicos en la PUJ - JPlanner

Autor(s):

Juan Sebastián Gómez

Nicolas Restrepo

Álvaro David Pongutá

Juan Felipe Hernández

Jesús Gilberto Pineda

UNDERGRADUATE FINAL PROJECT REPORT PERFORMED IN ORDER TO ACCOMPLISH
ONE OF THE REQUIREMENTS FOR THE SYSTEMS ENGINEERING DEGREE

Director

Jaime Andrés Pavlich

Juries of the Undergraduate Final Project

Anabel Montero Posada

Carlos Andres Parra

Website of Undergraduate Final Project

<https://pegasus.javeriana.edu.co/~CIS1830CP04/>

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
SYSTEMS ENGINEERING PROGRAM
BOGOTÁ, D.C.
Mayo2019

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
SYSTEMS ENGINEERING PROGRAM**

President of the Pontificia Universidad Javeriana

Jorge Humberto Peláez Piedrahita, S.J.

Dean of School of Engineering

Eng. Jorge Luis Sánchez Téllez

Head of the Systems Engineering Program

Mariela Josefina Curiel

Head of the Systems Engineering Department

Efrain Ortiz Pabon

Artículo 23 de la Resolución No. 1 de Junio de 1946

“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia”

AGRADECIMIENTOS

Queremos agradecer a nuestros padres los cuales fueron un gran apoyo durante todos estos años de estudio. Por otro lado, es importante agradecer a cada miembro de este equipo ya que gracias a su esfuerzo y entrega hoy podemos finalizar este trabajo de grado. A nuestros profesores por todo lo que nos dejaron en estos años de carrera, en especial a nuestro director Jaime Andrés Pavlich quien junto a nuestro equipo dio lo mayor de si para conseguir el mejor de los resultados.

CONTENIDO

I-	INTRODUCCIÓN	1
II-	DESCRIPCIÓN GENERAL	2
1.	OPORTUNIDAD, PROBLEMA	2
1.1.	<i>Contexto del problema.....</i>	<i>2</i>
1.2.	<i>Formulación del problema.....</i>	<i>2</i>
1.3.	<i>Propuesta de solución.....</i>	<i>3</i>
1.4.	<i>Justificación de la solución.....</i>	<i>3</i>
2.	DESCRIPCIÓN DEL PROYECTO	5
2.1.	<i>Objetivo general.....</i>	<i>5</i>
2.2.	<i>Objetivos específicos.....</i>	<i>6</i>
2.3.	<i>Entregables, Estándares utilizados y Justificación.....</i>	<i>6</i>
III-	CONTEXTO DEL PROYECTO.....	7
1.	CONTEXTO	7
2.	ANÁLISIS DEL CONTEXTO	8
IV-	ANÁLISIS DEL PROBLEMA.....	11
1.	REQUERIMIENTOS	12
	REQUISITOS DE INTERFACES EXTERNAS:	12
1.	<i>Interfaces con el usuario.....</i>	<i>12</i>
2.	<i>Interfaces con el software.....</i>	<i>12</i>
	CARACTERÍSTICAS DEL PRODUCTO SOFTWARE	13
	ATRIBUTOS DEL SISTEMA SOFTWARE	16
2.	RESTRICCIONES	22
3.	ESPECIFICACIÓN FUNCIONAL	23
V-	DISEÑO DE LA SOLUCIÓN.....	26
	VISTA DE PROCESOS DEL SISTEMA.....	36
VI-	DESARROLLO DE LA SOLUCIÓN	47
VII-	RESULTADOS.....	53
VIII-	CONCLUSIONES	55

1.	ANÁLISIS DE IMPACTO DEL PROYECTO	55
2.	CONCLUSIONES Y TRABAJO FUTURO	56
IX-	REFERENCIAS	57

RESUMEN

Este trabajo es la solución que el equipo propone para un problema que la DTI (Dirección de tecnologías de la información) propuso en una reunión en el año 2018. JPlanner es un sistema que cambiará el método de registro de clase en la Pontificia Universidad Javeriana. Actualmente, el sistema PeopleSoft presenta muchos problemas sobre los cuales se profundizará más adelante, JPlanner busca ser el sistema de estudiantes para estudiantes que solucione la mayoría de estos problemas brindando la mejor solución para la universidad. Se realizaron pruebas para medir la aceptación del sistema final, los resultados de estos estudios nos muestran la aceptación de nuestro sistema por parte de los estudiantes y nos ayudan brindando oportunidades de trabajo futuro para mejorar la aplicación.

I- INTRODUCCIÓN

A continuación se presenta el documento de reporte final del trabajo de grado correspondiente al aplicativo web JPlanner, el cual es un aplicativo web que brinda múltiples funcionalidades como lo son la búsqueda filtrada de materias, creación de varias opciones de horario en simultaneo, ver información detallada de las materias a inscribir, guardar horarios y seleccionar favoritos para ser usados en el momento de formalizar la inscripción de materias, generar horarios automáticamente basándose en filtros escogidos por el estudiante con la opción de incorporar bloqueos de franjas horarias en caso de que el estudiante así lo solicite.

El aplicativo web busca solucionar los problemas que presentan los estudiantes de la Pontificia Universidad Javeriana en el momento de inscribir materias, los problemas más comunes presentados por la comunidad Javeriana son:

- No es posible crear horarios en la aplicación actual, se limita a tan solo poseer un listado de materias.
- No es posible recibir sugerencias de horarios por parte de la aplicación.
- Aunque existen algunos filtros, no son los más adecuados y al momento de ver las opciones se vuelve engorroso.
- No se permite generar bloqueos a franjas horarias deseadas por el estudiante.
- Interfaz desarrollada sin pensar en el usuario final.

El presente documento busca hacer un resumen detallado de todos los factores importantes en la realización del sistema, así como mostrar los resultados obtenidos a las distintas pruebas elaboradas al sistema final, dar una idea del posible impacto que genera el resultado final y brindar conclusiones y posible trabajo futuro para la mejora o continuación del sistema desarrollado.

II- DESCRIPCIÓN GENERAL

1. Oportunidad, Problema

1.1. Contexto del problema

Cada periodo los estudiantes de la Pontificia Universidad Javeriana deben realizar la inscripción de las materias que cursarán en el siguiente semestre académico. Este proceso se realiza a través del sistema *PeopleSoft* el cual ofrece un carrito de compras en donde los estudiantes seleccionan las posibles materias que quieren cursar. Una vez inicia el periodo de inscripción de materias, el estudiante selecciona cuáles materias inscribirá y le envía una petición al sistema con el grupo de materias que desea cursar. De haber algún conflicto con varias materias que tienen horarios en común, el sistema no notifica al usuario antes de realizar la petición, pero sí impide la inscripción de cualquiera de las materias. De esta forma, no hay distintas alternativas de horario dentro del carrito de compras y no hay manera de crear parámetros para la planificación del horario (tales como días u horas libres).

1.2. Formulación del problema

Antes del periodo de inscripción de materias en las citas académicas, los estudiantes buscan las materias que deben inscribir, las cuales están incluidas en el plan de estudios de su carrera. Estas materias no se encuentran con mucha facilidad en el sistema de *PeopleSoft*. Las consultas arrojadas pueden ser muy extensas e inclusive erróneas respecto a las preferencias escogidas por un estudiante. Estas pueden incluir el código del curso, la facultad que la ofrece, la carrera específica, horarios, días a la semana, número de créditos, profesor que la dicta y algunos otros.

Además de no encontrar los cursos con facilidad, el sistema no ofrece recomendaciones respecto a posibles materias que se pueden cursar a partir de las anteriormente aprobadas, o la carrera que cursa actualmente un estudiante. Estos tienen como obligación revisar el pensum para encontrar esta información y saber con exactitud qué materias deben cursar durante ese semestre académico. Asimismo, el sistema no tiene en cuenta las restricciones de los cursos y admite la selección de estos hasta la inscripción, momento en que se notifica el por qué no se pudo realizar el respectivo procedimiento.

Finalmente, el sistema de *PeopleSoft* no ofrece ninguna herramienta gráfica durante el periodo anterior a la inscripción de materias, este se despliega únicamente al momento posterior a la inscripción. Esto es un problema para los estudiantes ya que tienen que hacer uso de herramientas gráficas adicionales para este proceso, y de esta forma percatar si existe algún tipo de conflicto de horario entre dos o más materias.

1.3. Propuesta de solución

Frente al servicio ofrecido por la plataforma actual de *PeopleSoft*, se planea desarrollar una aplicación web con diseño responsive (o diseño adaptativo – que se adapta a la pantalla del dispositivo donde se esté visualizando) con una interfaz más intuitiva, mejorando la experiencia del usuario. Principalmente, el desarrollo estará orientado a la visualización de la información y la interacción del usuario. Esta aplicación estará desacoplada del sistema de *PeopleSoft* y consumirá los servicios que expone el sistema para proveer las funcionalidades de inscripción de materias a los usuarios.

1.4. Justificación de la solución

En este apartado se realizará una comparativa que permita entender el por qué la solución propuesta es la más adecuada para la resolución del problema planteado. Para realizar esta comparativa se realizó una búsqueda de herramientas que se adaptarán a la solución propuesta por el equipo, tras investigar en distintos portales y artículos, el equipo tomo la decisión de realizar una comparativa entre las tres herramientas con mayor acogida al momento en el mercado y de mayor uso por parte de los programadores. A continuación se muestra la comparativa de las herramientas mencionadas anteriormente y la opción tomada por el equipo [21][22]

Angular7: es un framework de JavaScript utilizado para crear y mantener aplicaciones web. Su principal objetivo es el desarrollo mediante el modelo vista controlador (MVC), para que así tanto las pruebas como el desarrollo sean más fáciles y eficientes.

ReactJS: es una biblioteca de JavaScript enfocada en las interfaces de usuario. Está construida exclusivamente para el uso del patrón modelo vista controlador (MVC). Además, tiene varios beneficios enfocados en el performance, modularidad y en el flujo de datos y eventos.

VueJS: Es un framework enfocado en el consumo de interfaz de usuario, el cual es adaptable incrementalmente a otras bibliotecas en donde su biblioteca principal se enfoca solo en la capa de la vista.

A continuación, se presenta un cuadro comparativo entre las herramientas seleccionadas:

Tabla1. **Comparativo herramientas**

Herramientas/Frameworks	Ventajas	Desventajas

Angular7	<ul style="list-style-type: none">• Creación de aplicaciones web con gran interacción con el usuario• Gran velocidad de compilación gracias a la tecnología RXJS• Permite el flujo bidireccional de los datos. Esto permite un mayor desacople de la aplicación• MVVM es un modelo que permite a los desarrolladores trabajar desacopladamente sobre la misma aplicación desde diferentes módulos• Grandes compañías como Google, Youtube y Nike, hacen uso de este Framework Google	<ul style="list-style-type: none">• Migrar a nuevas versiones puede tener diversos problemas• Sintaxis con mayor complejidad que la de sus aplicaciones contiguas
ReactJS	<ul style="list-style-type: none">• Sintaxis poco compleja, esto lo hace muy fácil de aprender• Puede trabajar con alta carga gracias a la tecnología ES6/7• Es 100% código libre, lo que permite grandes actualizaciones por parte de los colaboradores• Migrar de una versión anterior suele ser muy fácil	<ul style="list-style-type: none">• No es una herramienta que posee fuertes opiniones o comentarios de programadores. Lo que hace que los mismos tengan muchas más opciones de herramientas a evaluar• No posee un fácil acoplamiento con distintos frameworks como el MVC• Uso de JSX lo que lo distancia de HTML y lo hace más complicado para su aprendizaje• El flujo de datos es unidireccional
Vue.js	<ul style="list-style-type: none">• Posee una interfaz amigable que puede ayudar a optimizar bloques de HTML• Es flexible y de poco peso. Esto permite un gran desempeño	<ul style="list-style-type: none">• Tiene un mercado en una fase inicial, lo que no permite que tenga muchos recursos a disposición• Poca documentación, y la mayoría de la misma

	<ul style="list-style-type: none"> • Por su diseño muy similar a otros Frameworks, como Angular, su adaptabilidad es muy sencilla 	<p>se encuentra en Ingles. Esto dificulta su aprendizaje</p> <ul style="list-style-type: none"> • Poco tiempo en el mercado aún en fase inicial
--	--	--

Con dicha comparativa el equipo justifica el uso de Angular7 debido a la gran documentación que se posee, la bidireccionalidad en el flujo de datos, el respaldo de grandes compañías y programadores a la herramienta, alta interactividad con el usuario, lo que permite al equipo diseñar una aplicación amigable para los estudiantes de la Pontificia Universidad Javeriana.

Además, esta permite el trabajo desacoplado tanto de la solución como de sus componentes internos, lo que minimiza la probabilidad de fallas y cumplir con los objetivos deseados. Adicionalmente, debido al previo conocimiento de la tecnología por todo el equipo la curva de aprendizaje fue mucho menos larga que con las demás tecnologías.

2. Descripción del proyecto

Se desarrolla una solución a la problemática descrita en la sección **1.1.2 Formulación del Problema** mediante una aplicación web caracterizada por tener una interfaz intuitiva, que facilitará a los estudiantes de la Pontificia Universidad Javeriana planificar su horario con base a las asignaturas que tienen que cursar en un determinado semestre. Esto se hará mediante el consumo de servicios web provistos por *PeopleSoft*, haciendo uso de ciertas funcionalidades como lo son las credenciales del usuario, la información de las materias a inscribir, entre otras. Es importante resaltar que también el aplicativo deberá proveer ciertos servicios web que serán consumidos por *PeopleSoft*, para así poder concretar la inscripción de materias en el carrito de compras de dicho sistema. El fin de esta solución es facilitar la inscripción de materias, sugiriendo así distintos tipos de horarios, que tengan en cuenta ciertos datos de enriquecimiento ingresados por el usuario.

2.1. Objetivo general

Desarrollar una aplicación web donde los estudiantes de la Pontificia Universidad Javeriana puedan planear versiones tentativas de su horario académico.

2.2. Objetivos específicos

- Realizar el levantamiento de requerimientos
- Diseñar al menos dos alternativas de mockups para la interfaz del aplicativo web
- Consumir los servicios de *PeopleSoft* expuestos por la DTI
- Realizar un prototipo del sistema de inscripción de materias

2.3. Entregables, Estándares utilizados y Justificación

Tabla2. Entregables

Entregable	Estándares asociados	Justificación
Pruebas	QUIS (Questionnaire for user interface satisfaction)	Define el vocabulario, procesos, documentación, técnicas y un modelo de evaluación de procesos para pruebas de usabilidad.
Código fuente	Estandares en Spring JavaDoc [6], estandares Angular7 [7]	Define métodos de documentación de funciones y variables
SPMP	1058-1998 - IEEE Standard for Software. Project Management Plan.	Define el plan de proyecto propuesta para el desarrollo del producto final.
SRS	830-1998 - IEEE Recommended practice for software requirements specifications.	Se especifican los requisitos de software que tendrá el producto final, la documentación de los acuerdos entre el cliente y las interfaces que tendrá
SDD	1016-2009 - IEEE Standard for information technology, systems design, software design description	Describe el diseño y la arquitectura del sistema, el diseño de las interfaces internas y externas y el diseño de procesos.

III- CONTEXTO DEL PROYECTO

1. Contexto

En los últimos años, el desarrollo web basado en el diseño adaptativo (responsive design) ha sido clave para mejorar la experiencia de los usuarios en el internet. Debido a que, al acceder a una página web desde diferentes dispositivos, cambia el aspecto de esta dependiendo de las dimensiones de la pantalla desde la cual se está visualizando.

El principal objetivo de una página web adaptativa es aumentar el número de usuarios que pueden acceder al sitio e interactuar en él sin sentirse incómodos, donde todas las características de la página estén al alcance del usuario y el contenido ocupe una posición apropiada dentro del espacio de visualización disponible. Es importante tener en cuenta que un diseño adaptativo da la posibilidad a una organización de ya no tener una versión móvil y otra de escritorio de su negocio, sino un único sitio web.

“El diseño responsive trata de proporcionar al usuario la mejor experiencia posible, no importa que el dispositivo que estén utilizando. Se trata de hacer más fácil la navegación al usuario, explorar y comprender para que puedan obtener exactamente lo que quieren de un sitio” [1]

Los elementos más importantes dentro del desarrollo web adaptativo son:

- **Diseño fluido:** utiliza una grilla flexible la cual asegura que un sitio web pueda escalar la organización del contenido en diferentes resoluciones.
- **Imágenes que funcionen en un contexto flexible:** imágenes que se adaptan dependiendo de los límites del sitio web.
- **Media queries:** los cuales optimizan el diseño para diferentes contextos de visualización y que ayudan a la corrección de errores que ocurren en los diferentes rangos de resolución. Generalmente los media queries definen los puntos de quiebre en la resolución donde el sitio web cambia la vista del contenido.

Para propósitos del desarrollo de JPlanner, se desarrollará una aplicación web adaptativa con tres puntos de quiebre diferentes, debido a que son los tipos de dispositivos más comunes en el mercado:

- Para celulares
- Para tabletas

- Para dispositivos de escritorio

Un framework para aplicaciones web permite el desarrollo de sitios web dinámicos, enfocándose en la interacción del usuario con la página web, aplicando todos los conceptos de un diseño web adaptativo.

“Un framework para aplicaciones web es una herramienta de software que provee una forma de construir y ejecutar aplicaciones web. Como resultado, no es necesario escribir código propio ni gastar tiempo posibles errores de cálculo y errores.” [3]

Angular es un framework basado en JavaScript que permite crear aplicaciones web enfocado en el desarrollo web basado en interfaces sin la necesidad de usar otros plugins o frameworks.

Debido a su extensa documentación, Angular será el framework que se utilizará para el desarrollo de la aplicación web *JPlanner*.

2. Análisis del contexto

A continuación, se describirán todos los trabajos relacionados con nuestra propuesta de proyecto:

College Scheduler

Este trabajo desarrollado por Civitas Learning, es un planeador de clases el cual facilita elaborar un horario de manera rápida y eficaz, ya que agiliza el proceso de inscripción de cursos mediante esta aplicación web. Civitas Learning *“es una plataforma de éxito estudiantil que aprovecha los datos únicos de cada institución para encontrar y distribuir el camino más claro para mejorar los resultados de la educación superior. Lo hacemos a través de nuestra tecnología de vanguardia, pensamiento de diseño y ciencia de datos para lograr nuestro objetivo de ayudar a un millón de estudiantes más” [4].*

El college scheduler es utilizado por esta plataforma para mejorar la experiencia del estudiante a la hora de realizar una planeación de horarios. Esto se hace mediante la captura de información de obligaciones externas y tiempos de aprendizaje preferidos por la persona para que así el sistema pueda elaborar el mejor horario posible para dicho individuo. Este proceso se hace de manera satisfactoria en cuanto a la captura de preferencias de usuario para el aprendizaje, sin embargo, no es muy intuitivo, pero tiene una muy buena interfaz gráfica ya que combina bien los colores y maneja botones agradables a la vista. Otra falencia de este sistema es que toca ir añadiendo cursos al horario, el problema está en que al añadir un curso es necesario añadir toda la información ya que no se encuentra en una base de datos.

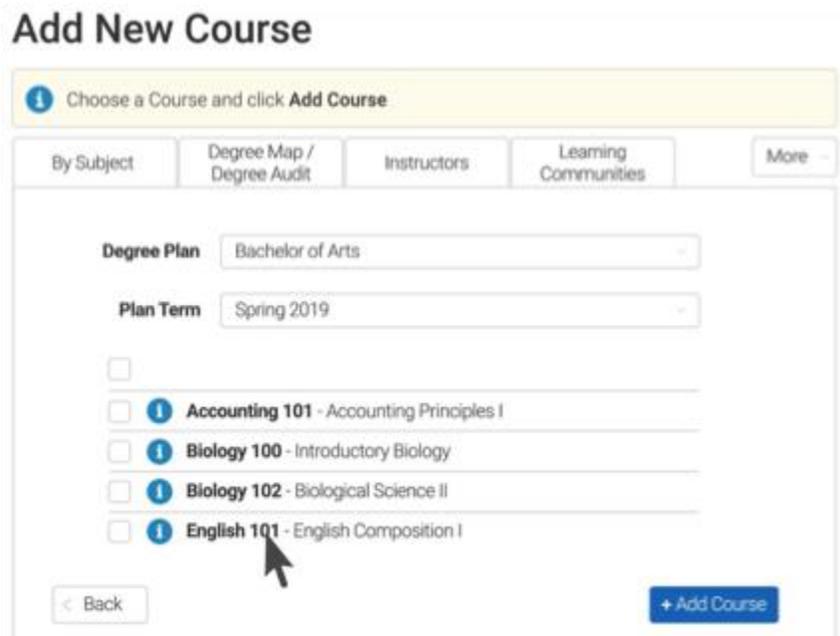


Imagen1. CS parte 1

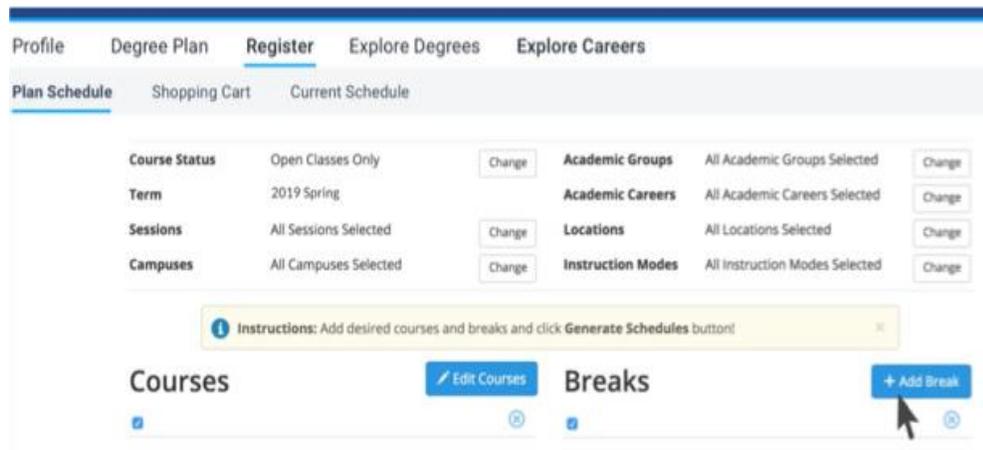


Imagen2. CS parte 2

Visual Schedule Builder

Este trabajo desarrollado por la Universidad de York, es un planificador de programas de cursos para los estudiantes, el cual está muy bien realizado debido a que dispone una base de datos con toda la información de todos los cursos que ofrece la universidad. Una desventaja de esta aplicación web es la

usabilidad de la interfaz gráfica, ya que no es nada intuitiva y además no es agradable a la vista.

Respecto a la funcionalidad es bastante buena, ya que cumple con el objetivo de facilitar la planificación de horarios, resolviendo así, la problemática que tienen los estudiantes a la hora de realizar una planeación de los cursos que tomarán en determinado semestre. Además, genera todos los posibles horarios teniendo en cuenta las horas en las cuales los cursos son dictados, generando así, varias opciones para que el estudiante escoja el que más se adapte al tiempo de aprendizaje deseado por el mismo y permita la selección de un resultado como favorito.

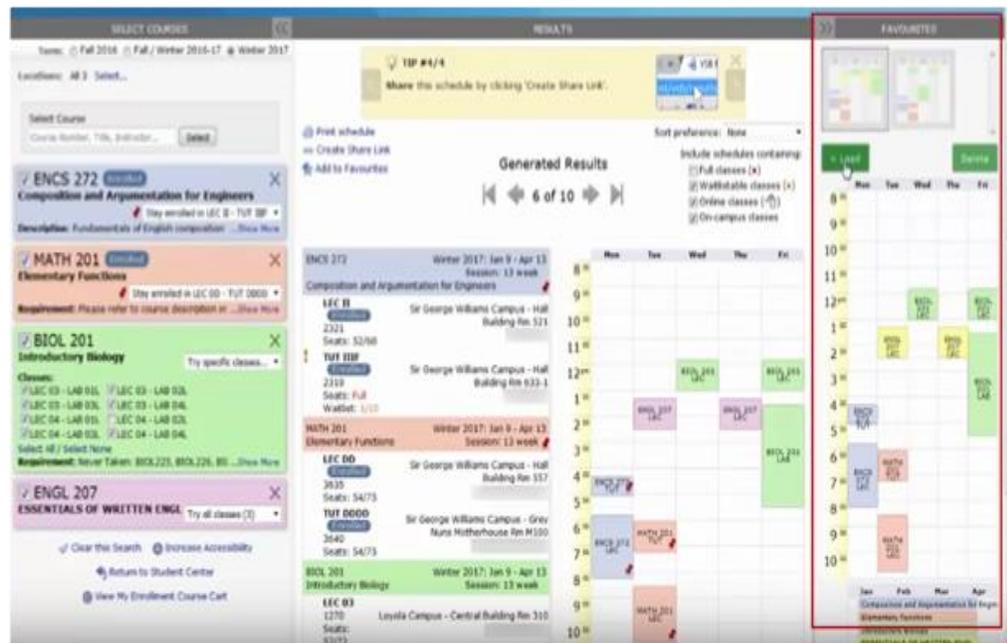


Imagen3. VSB

Free Class Schedule Maker Online

Este trabajo desarrollado por una comunidad llamada StudenGizmo, es un generador de horarios en línea muy sencillo, que sirve para visualizar de manera gráfica el horario de trabajo o de la universidad, dependiendo del usuario que lo utilice. Es muy fácil de usar, ya que su funcionalidad también es sencilla, algo muy interesante de esta, es que el horario tiene en cuenta las horas y los días de los cursos que sean ingresados, e ir creciendo dinámicamente. Además, el resultado final puede ser guardado como imagen o imprimirse directamente.

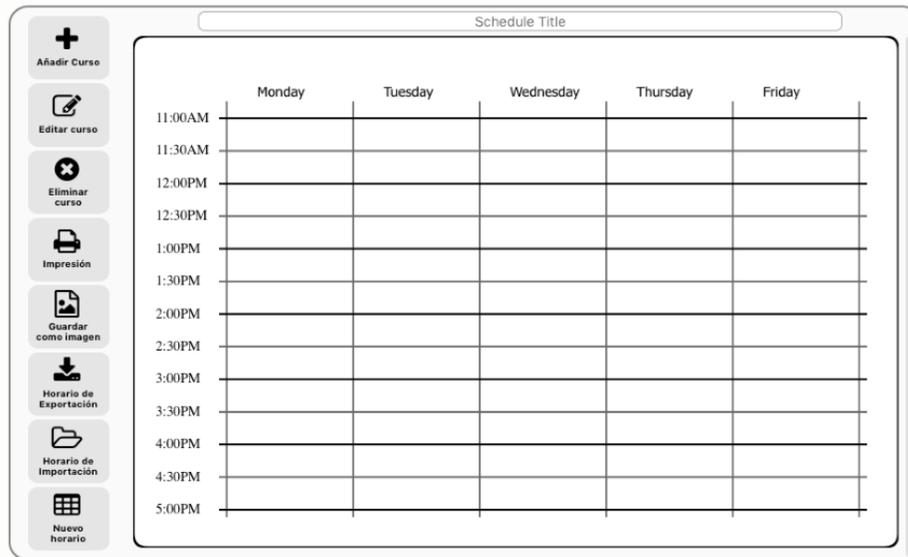


Imagen4. FCS

IV- ANÁLISIS DEL PROBLEMA

En la presente sección se van a exponer los principales aspectos correspondientes a la especificación de requisitos tanto en aspecto funcional como no funcional, adicionalmente, se tomarán aspectos importantes del diseño como lo son las especificaciones de las funcionalidades del sistema, las interfaces que van a interactuar con el usuario y el sistema, los atributos más importantes en cuanto a usabilidad, portabilidad, disponibilidad y confiabilidad. Lo que permite dar al lector una base sólida con la cual entender el alcance adecuado del proyecto.

Esta sección es muy importante ya que se elabora con el propósito de documentar de una forma más robusta los aspectos primordiales que permiten elaborar el alcance adecuado del proyecto. Contemplando las principales funcionalidades del sistema, las restricciones en cuanto a hardware y software que se poseen, así como el levantamiento final de los requisitos estipulados por la DTI y la formulación de los casos de uso que conforman el sistema.

1. Requerimientos

Para esta sección se manejarán tres tipos de prioridad para los diferentes requisitos que debe tener la aplicación:

- **ALTA:** La característica tiene que estar en la entrega del prototipo.
- **MEDIA:** La característica debería estar implementada en la entrega del prototipo, pero si no lo está no generará conflictos con las características que tienen que estar.
- **BAJA:** La característica podría estar o no en la entrega del prototipo.

Es importante mencionar que los requerimientos considerados con prioridad alta son aquellos que se encuentran como alcance del proyecto, ya que ayudan en la definición del prototipo final.

Requisitos de interfaces externas:

Esta sección se busca simplemente profundizar en el contenido que debe tenerse en cuenta para definir los requisitos de las interfaces externas del sistema.

1. Interfaces con el usuario

La aplicación JPlanner permitirá una interacción táctil en los dispositivos móviles y tablets, en cambio en los computadores, se llevará a cabo una interacción convencional con el mouse y el teclado. Ambas interacciones se llevarán a cabo en una pantalla digital y funcionarán como interfaz de comunicación entre el usuario y el software. Se espera que el acceso a la aplicación sea a través de diferentes plataformas y es por esto que la GUI debe ser adaptable a los diferentes dispositivos. Respecto a la distribución de elementos en la pantalla, se diseñarán pantallas simples donde las dimensiones de los botones sean lo suficientemente grandes de forma que ningún usuario tenga inconvenientes a la hora de interactuar con el sistema. Además, se tendrá en cuenta la escala de la letra para que los usuarios puedan reconocer el texto desplegado en pantalla.

2. Interfaces con el software

Para que el sistema pueda cumplir con sus funcionalidades principales, se comunica con otros productos descritos a continuación:

Producto	Descripción	Propósito de uso	Versión	Fuente
----------	-------------	------------------	---------	--------

de software				
PeopleSoft	Las aplicaciones PeopleSoft de Oracle están diseñadas para satisfacer los requisitos empresariales más complejos. Brindan soluciones empresariales y sectoriales integrales, lo que permite a las organizaciones aumentar su productividad, acelerar el rendimiento de negocio y brindan un costo de propiedad reducido.	A partir del SIU (Sistema de Información Universitaria) los estudiantes podrán ingresar a JPlanner y el SIU se encargará de enviar las credenciales del usuario, de tal modo que no se tenga que autenticar. Además, PeopleSoft expondrá los servicios web necesarios, para la gestión de la información de asignaturas localizada en la base de datos de la Pontificia Universidad Javeriana.	9.2	https://www.oracle.com/lad/products/applications/peoplesoft-enterprise/overview/
Javemóvil	Aplicación dispositivos móviles de la Javeriana, con la que los estudiantes pueden saber de sus clases, salones y edificios; contactar profesores y compañeros; mantenerse al tanto de las noticias y eventos, entre otras.	A partir de la aplicación los estudiantes podrán ingresar a JPlanner y Javemóvil se encargará de enviar las credenciales del usuario, de tal modo que no se tenga que autenticar.		https://www.javeriana.edu.co/javemovil/

Tabla 3. – interfaces con el software

Características del producto software

En este apartado se van a detallar todas las características necesarias a tener en el sistema, es decir, toda característica que será detallada en las tablas siguientes cumple como un requisito de prioridad alta y por esto deben ser implementadas.

# Requisito	1	Buscar materias	El sistema debe permitir buscar una materia que se encuentre en PeopleSoft
Descripción	El usuario ingresa en el buscador los parámetros de las asignaturas que desea buscar. Los parámetros que se pueden ingresar son los que actualmente se encuentran en el servicio de buscar materias de PeopleSoft.		
Razón	Los estudiantes deben poder filtrar las materias que desean ver según su plan académico personal.		
Autor	JPlanner		
Prioridad	ALTA		
Versión	1	Fecha	11 de noviembre de 2018

Tabla 4. – Características del software 1

# Requisito	2	Agregar materia al horario	El sistema debe permitir agregar una materia de los resultados de búsqueda al horario
Descripción	El usuario selecciona una de las materias de los resultados de búsqueda y la ubica en uno de sus horarios		
Razón	Los estudiantes deben poder realizar versiones tentativas de su horario		
Autor	JPlanner		
Prioridad	ALTA		
Versión	1	Fecha	11 de noviembre de 2018

Tabla 5. – Características del software 2

# Requisito	3	Crear horario	El sistema debe permitir crear un horario
Descripción	El usuario puede crear uno o más horarios en la aplicación, a los cuales puede agregarle materias		
Razón	Los estudiantes deben poder realizar versiones tentativas de su horario		
Autor	JPlanner		
Prioridad	ALTA		
Versión	1	Fecha	11 de noviembre de 2018

Tabla 6. – Características del software 3

# Requisito	4	Guardar horario	El sistema debe permitir guardar un horario
Descripción	El usuario puede guardar sus horarios con diferentes materias cada uno		
Razón	Los estudiantes deben poder realizar versiones tentativas de su horario		
Autor	JPlanner		
Prioridad	ALTA		
Versión	1	Fecha	11 de noviembre de 2018

Tabla 7. – Características del software 4

# Requisito	5	Eliminar materia del horario	El sistema debe permitir eliminar una materia de un horario
Descripción	El usuario puede escoger las materias que quiere eliminar de sus horarios		
Razón	Los estudiantes deben poder realizar versiones tentativas de su horario		
Autor	JPlanner		
Prioridad	ALTA		
Versión	1	Fecha	11 de noviembre de 2018

Tabla 8. – Características del software 5

# Requisito	6	Eliminar horario	El sistema debe permitir eliminar un horario
Descripción	El usuario puede escoger las versiones tentativas de los horarios que desea guardar o eliminar del sistema		
Razón	Los estudiantes deben poder realizar versiones tentativas de su horario		
Autor	JPlanner		
Prioridad	ALTA		
Versión	1	Fecha	11 de noviembre de 2018

Tabla 9. – Características del software 6

# Requisito	7	Inscribir materias	El sistema debe permitir inscribir un horario
Descripción	El usuario puede inscribir las materias que tiene en una de sus versiones tentativas de horarios		
Razón	Los estudiantes deben poder inscribir materias		
Autor	JPlanner		
Prioridad	ALTA		
Versión	1	Fecha	11 de noviembre de 2018

Tabla 10. características del software 7

# Requisito	8	Ver detalles de materia	El sistema debe permitir una visualización detallada de las materias
Descripción	El usuario debe poder ver la información de cada materia, la información que podrán visualizar será la misma que actualmente se puede ver en PeopleSoft		
Razón	Los estudiantes deben poder conocer a fondo las materias que pueden inscribir en su horario		
Autor	JPlanner		
Prioridad	ALTA		
Versión	1	Fecha	11 de noviembre de 2018

Tabla 11. – Características del software 8

# Requisito	9	Ver detalles de materia	El sistema debe generar una recomendación de horario
Descripción	El sistema debe generarle una versión tentativa del horario al usuario según los parámetros que el usuario le ingrese		
Razón	Los estudiantes deben poder realizar versiones tentativas de su horario		
Autor	JPlanner		
Prioridad	ALTA		
Versión	1	Fecha	11 de noviembre de 2018

Tabla 12. – Características del software 9

Atributos del sistema software

En esta sección se describirán los requisitos no funcionales que se deberán implementar para que el aplicativo web JPlanner funcione de manera óptima teniendo en cuenta las características solicitadas por el *product owner* (La dirección de tecnologías e información de la Universidad Javeriana). Dichos requisitos se centran en los atributos de usabilidad, disponibilidad, portabilidad y confiabilidad.

Usabilidad

Para nuestro proyecto este es el atributo de calidad más importante de todos. Uno de los problemas que tiene el portal usado actualmente para inscribir materias en la Universidad Javeriana es precisamente el uso de interfaces gráficas complicadas y muy poco interactivas. Es por esto por lo que se deben implementar interfaces que sean no solo intuitivas, sino también dinámicas y atractivas para el usuario, es por esto que se definen los siguientes requisitos:

# Requisito	10	Tipo de requisito	NF
Descripción	El sistema debe permitir agregar y eliminar materias del horario en menos de 4 clicks.		
Razón	Actualmente el sistema para inscribir materias supone un proceso muy engorroso y tedioso a la hora de agregar o eliminar una materia.		
Autor	JPlanner.		
Criterio de medición	Se utilizarán focos de estudio con el prototipo de la aplicación y con la retroalimentación de ellos se medirá si fue o no cumplido este requisito.		
Prioridad	Alta		
Versión	1.0	Fecha	10/11/2018

Tabla 13. – Usabilidad 1

# Requisito	11	Tipo de requisito	NF
Descripción	El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 1 hora.		
Razón	El sistema debe ser intuitivo de usar, con elementos gráficos que faciliten el uso del aplicativo. No se debe gastar mayor tiempo en capacitaciones para el usuario del producto.		
Autor	JPlanner.		
Criterio de medición	Con un foco de estudio, y midiendo su tiempo de aprendizaje podremos medir si fue o no cumplido este requisito.		
Prioridad	Alta		
Versión	1.0	Fecha	10/11/18

Tabla 14. – Usabilidad 2

# Requisito	12	Tipo de requisito	NF
Descripción	El sistema debe presentar mensajes de error claros, evitando ambigüedades.		
Razón	Si un estudiante no puede inscribir una materia, el mensaje de error debe ser claro explicando los motivos de porque no se pudo, de esta manera el usuario final podrá tomar acciones correctivas.		
Autor	JPlanner.		
Criterio de medición	En conjunto con la DTI evaluaremos todos los casos de errores al intentar inscribir materias y se definirán los mensajes que se deben mostrar, al abarcarlos todos, estamos cumpliendo con el requisito.		
Prioridad	Media		
Versión	1.0	Fecha	10/11/2018

Tabla 15. – Usabilidad 3

# Requisito	13	Tipo de requisito	NF
Descripción	El sistema debe contar con un diseño responsivo.		
Razón	El usuario final podrá acceder a la aplicación desde un teléfono celular, un computador de escritorio o una Tablet y en cada uno de estos se debe visualizar de manera correcta y consistente los elementos visuales de la aplicación.		
Autor	JPlanner.		
Criterio de medición	La aplicación se probará en múltiples dispositivos para garantizar el correcto cumplimiento de este requisito.		
Prioridad	Alta.		
Versión	1.0	Fecha	10/11/2018

Tabla 16. – Usabilidad 4

# Requisito	14	Tipo de requisito	NF
Descripción	El sistema debe contar con un componente de ayuda tutorial.		
Razón	El usuario debe encontrar dentro de la aplicación una sección de ayuda donde se le explique claramente cómo utilizar la aplicación.		
Autor	JPlanner.		
Criterio de medición	Número de veces que el componente de ayuda es utilizado.		
Prioridad	Media.		
Versión	1.0	Fecha	10/11/2018

Tabla 17. – Usabilidad 5

Disponibilidad

Es necesario para la aplicación una alta disponibilidad para que el usuario pueda planear un horario académico. Se debe aclarar que a pesar de que el aplicativo estará conectado con los servicios de PeopleSoft, la disponibilidad de JPlanner no se verá afectada por la disponibilidad del sistema de PeopleSoft ya que son módulos completamente independientes. A continuación, se documentarán los requisitos necesarios para cubrir la disponibilidad del aplicativo web JPlanner.

# Requisito	15	Tipo de requisito	NF
Descripción	El sistema debe tener una disponibilidad del 99% de las veces en que un usuario intente accederlo.		
Razón	Al ser una aplicación independiente al sistema actual de PeopleSoft, debe estar disponible la mayor cantidad de tiempo posible, de esta manera el usuario final puede hacer su horario sin que este proceso se vea entorpecido por la disponibilidad del sistema PeopleSoft, el cual no está disponible antes de la cita de inscripción del estudiante.		
Autor	JPlanner.		
Criterio de medición	Porcentaje de disponibilidad de JPlanner para los usuarios.		
Prioridad	Media.		
Versión	1.0	Fecha	10/11/2018

Tabla 18. – Disponibilidad 1

# Requisito	16	Tipo de requisito	NF
Descripción	El tiempo para que la aplicación inicie no debe ser mayor a un minuto.		
Razón	Independientemente del dispositivo desde el que el usuario inicie la aplicación, esta debe estar inicializada en máximo 1 minuto.		
Autor	JPlanner.		
Criterio de medición	Tiempo en el cual la aplicación es iniciada.		
Prioridad	Media.		
Versión	1.0	Fecha	10/11/2018

Tabla 19. – Disponibilidad 2

# Requisito	17	Tipo de requisito	NF
Descripción	El tiempo para reiniciar el sistema no debe ser mayor a 10 minutos.		
Razón	Si ocurre algún fallo y es necesario reiniciar los servidores de la aplicación, no debe tomar más de 10 minutos.		
Autor	JPlanner.		
Criterio de medición	Tiempo en el cual se reinicia el sistema.		
Prioridad	Media.		

Versión	1.0	Fecha	10/11/2018
----------------	-----	--------------	------------

Tabla 20. – Disponibilidad 3

# Requisito	18	Tipo de requisito	NF
Descripción	La probabilidad de errores del sistema no puede ser mayor al 1%.		
Razón	Cuando un usuario inicia el sistema, las probabilidades de fallo deben ser prácticamente inexistentes		
Autor	JPlanner.		
Criterio de medición	Porcentaje de fallos al iniciar la aplicación utilizando un foco de estudio.		
Prioridad	Media.		
Versión	1.0	Fecha	10/11/2018

Tabla 21. – Disponibilidad 4

Portabilidad

La portabilidad es uno de los atributos de calidad más relevantes en el proyecto del aplicativo JPlanner, cabe la pena resaltar el hecho de que para el equipo la portabilidad es entendida como la funcionalidad del producto en diferentes dispositivos. Debido a que los estudiantes en varias ocasiones realizan el proceso de inscripción de materias a través de distintos dispositivos móviles, es necesario soportar los mismos, los requisitos que se deben abarcar para garantizar este atributo de calidad son:

# Requisito	19	Tipo de requisito	NF
Descripción	El sistema debe funcionar correctamente en buscadores web desde dispositivos de escritorio.		
Razón	La aplicación podrá ser accedida desde las versiones más recientes de los buscadores: Opera, Microsoft Edge, Google Chrome y Firefox.		
Autor	JPlanner.		
Criterio de medición	Si se puede acceder desde los navegadores web mencionados anteriormente o no.		
Prioridad	Alta.		
Versión	1.0	Fecha	11/11/2018

Tabla 22. – Portabilidad 1

# Requisito	20	Tipo de requisito	NF
Descripción	El sistema debe funcionar correctamente desde dispositivos Android desde la versión 6 de este sistema operativo.		

Razón	Se dará soporte a dispositivos Android desde su versión 6 hasta la más reciente.		
Autor	JPlanner.		
Criterio de medición	Si se puede acceder desde dispositivos Android 6 y superiores o no.		
Prioridad	Alta.		
Versión	1.0	Fecha	11/11/2018

Tabla 23.-- Portabilidad 2

# Requisito	21	Tipo de requisito	NF
Descripción	El sistema debe funcionar correctamente desde dispositivos IOS desde la versión 9 de este sistema operativo.		
Razón	Se dará soporte a dispositivos IOS desde su versión 9 hasta la más reciente.		
Autor	JPlanner.		
Criterio de medición	Si se puede acceder desde dispositivos IOS 9 y superiores o no.		
Prioridad	Alta.		
Versión	1.0	Fecha	11/11/2018

Tabla 24. – Portabilidad 3

Confiabilidad

Los requisitos para la confiabilidad en la aplicación JPlanner son:

# Requisito	22	Tipo de requisito	NF
Descripción	El sistema debe soportar 1200 usuarios concurrentes		
Razón	En conjunto con la DTI se determinó que este es el número máximo de usuarios concurrentes que utilizarán la aplicación.		
Autor	JPlanner.		
Criterio de medición	Número de transacciones concurrentes que soporta el aplicativo.		
Prioridad	Alta.		
Versión	1.0	Fecha	11/11/2018

Tabla 25. – Confiabilidad 1

# Requisito	23	Tipo de requisito	NF
Descripción	El sistema debe crear una copia de la base de datos cada 24 horas.		

Razón	Se debe prevenir cualquier fallo y para esto un respaldo automático de la base de datos debe ser realizado cada 24 horas.		
Autor	JPlanner.		
Criterio de medición	La base de datos es respaldada cada 24 horas.		
Prioridad	Alta.		
Versión	1.0	Fecha	11/11/2018

Tabla 26. – Confiabilidad 2

# Re-quisito	24	Tipo de requisito	NF
Des-crip-ción	La base de datos debe soportar 1200 conexiones abiertas haciendo operación de lectura.		
Razón	El número de conexiones abiertas hacia la base de datos es correspondiente al número de usuarios concurrentes que debe soportar la aplicación, por lo tanto, la base de datos debe soportar 1200 conexiones abiertas haciendo operaciones de lectura.		
Autor	JPlanner.		
Criterio de medición	La base de datos soporta 1200 conexiones concurrentes.		
Prioridad	Media.		
Versión	1.0	Fecha	11/11/2018

Tabla 27. – Confiabilidad 3

2. Restricciones

En esta sección se describen las restricciones generales, de software y hardware para el correcto desarrollo de JPlanner, todas las siguientes restricciones fueron definidas por todo el equipo mediante la definición del tipo de producto que se va a desarrollar y además por la identificación de las diferentes habilidades que cada integrante del equipo tiene, para que de esta manera se pueda maximizar el uso de dichas habilidades para poder asegurar un producto con una buena calidad teniendo en cuenta los requisitos. A partir de todo esto se generaron distintas restricciones de diseño que permiten cumplir con el alcance del producto las cuales son descritas a continuación:

Restricciones generales:

- La persona que utilice la aplicación debe tener conexión a internet.
- La aplicación deberá poder ser visualizada desde resoluciones de 360px de ancho para celulares, 768px de ancho para tabletas y 1280px de ancho para computadores de escritorio.
- La aplicación deberá consumir servicios web expuestos por la plataforma PeopleSoft utilizada por la Universidad Javeriana, sin embargo, la información que

se expondrá en los servicios será de acuerdo con las necesidades que presente la aplicación.

- La aplicación web deberá adherirse a los métodos de autenticación ofrecidos por la universidad, se debe implementar el protocolo que permita conectar y enviar datos con el sistema actual de SingleSignOn de Oracle el cual es utilizado por la DTI

Restricciones de hardware:

- El servidor deberá soportar 1200 conexiones de usuarios en concurrente

3. Especificación funcional

En esta sección se plantean las funcionalidades generales del sistema, con el objetivo de realizar la especificación del alcance del proyecto de una forma más precisa. A continuación, se explicarán estas funcionalidades mencionadas, así como la relación con los casos de uso que cada una de ellas posee. El diagrama de casos va a ser mostrado a continuación:

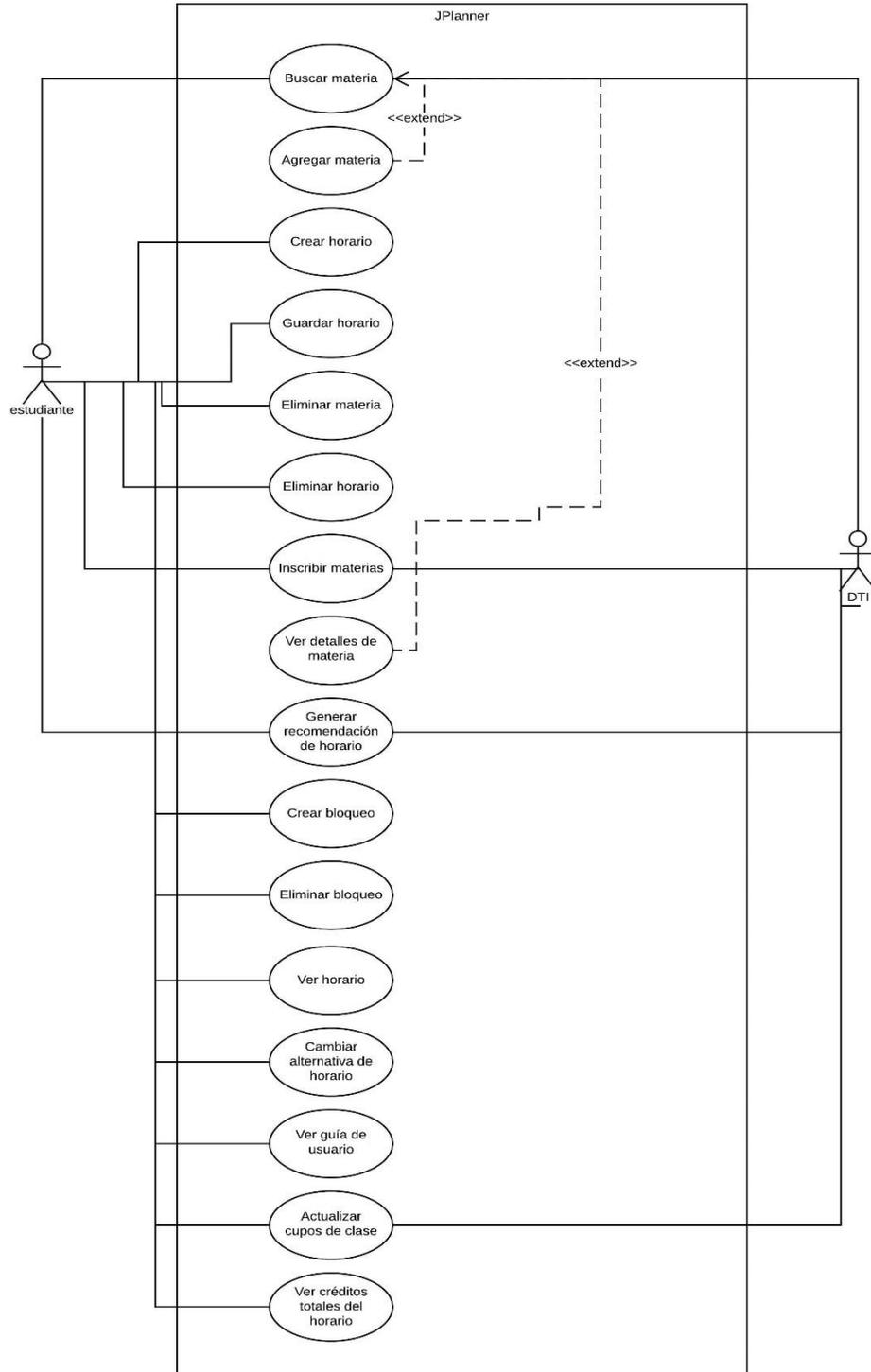


Imagen5. Diagrama casos de uso

Búsqueda de materias: Este requisito es importante ya que los estudiantes contarán con la función de búsqueda de materias la cual les permite buscar datos detallados de las materias deseadas a través de filtros y agregar materias a sus distintos horarios.

Casos de uso implicados:

- o Buscar materia
- o Agregar materia
- o Ver detalles de materia

Creación de horarios: Con la funcionalidad de este requisito los estudiantes pueden crear sus horarios académicos de forma manual, con la posibilidad de seleccionar que materias y en que horarios desean ingresarlas, además, teniendo la posibilidad de crear variadas opciones de horarios y guardarlos para su posterior uso o modificación.

Casos de uso implicados:

- o Buscar materia
- o Agregar materia
- o Crear Horario
- o Guardar horario
- o Ver Horario
- o Crear Bloqueo

Eliminar horarios: Este requisito permite a los usuarios la eliminación bien sea de una opción completa de horario o de las materias que lo componen, con el fin de realizar modificaciones a los existentes.

Casos de uso implicados:

- o Eliminar materia
- o Eliminar Horario
- o Eliminar Bloqueo

Inscripción de materias: A través de esta funcionalidad y/o requisito el estudiante podrá inscribir el horario deseado entre las diferentes alternativas creadas y de esta manera formalizando su inscripción de materias académica.

Casos de uso implicados:

- o Inscribir materias

Generación de horarios: Con este importante requisito los estudiantes podrán generar automáticamente diferentes opciones de horario, con la utilización de filtros que permiten entre otras cosas establecer bloqueos a franjas horarias en las cuales no se desea tener clases durante la semana.

Casos de uso implicados:

- o Generar recomendación de horario
- o Buscar materia
- o Agregar materia
- o Ver horario

Es importante mencionar que el actor DTI es aquel que expone servicios web, los cuales proveen la información real necesaria para el llenado de datos en el sistema, de esta manera interviniendo en distintas funcionalidades que posee JPlanner.

V- DISEÑO DE LA SOLUCIÓN

La presente sección contiene una descripción detallada de la arquitectura del sistema, en sus tres vistas física, lógica y de procesos. A su vez, posee el diseño detallado utilizado en la construcción del sistema, el detalle de la arquitectura permite al desarrollador entender con un nivel alto de abstracción como fue desarrollado el sistema y que estructura lo compone. Con el diseño se busca dar a conocer como es el comportamiento del sistema a más bajo nivel, explicando sus interfaces, comportamiento, su estructura, entre otros.

Arquitectura

Vista lógica del sistema

Un diagrama de componentes representa las dependencias entre componentes software, incluyendo componentes de código fuente, componentes del código binario, y componentes ejecutables. Por otro lado, permite a los desarrolladores una visión general de la arquitectura implementada en el sistema. Esta sección comprende el diagrama mencionado junto con la descripción de cada uno de sus componentes [11]. El diagrama puede ser encontrado a continuación

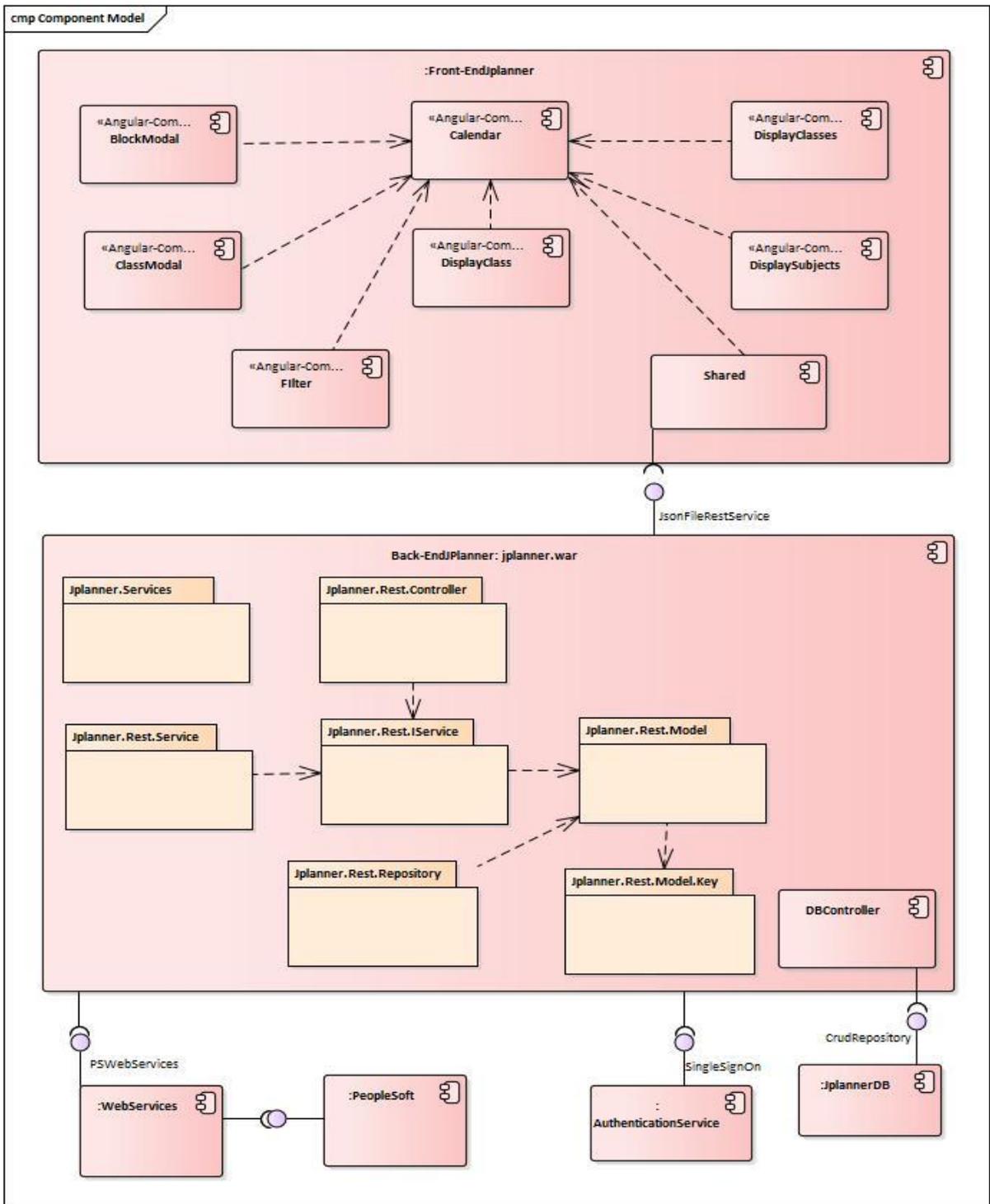


Imagen6.Diagrama Componentes

Para el desarrollo de nuestra plataforma, se utilizó el modelo MVC (Modelo Vista Controlador), Según Uriel Hernandez en su artículo MVC (Model, View, Controller) Explicado. *“es un patrón de arquitectura de software que, utilizando 3 componentes, las vistas, los modelos y los controladores, separa la lógica de la aplicación de la lógica de la vista en el sistema”*. Este patrón fue utilizado ya que permite la separación de la aplicación en componentes lo que permite un manejo adecuado de del código del sistema debido a que se pueden manejar los cambios fácilmente, y es considerada como una mejor práctica de programación por el orden que este brinda al código.

A continuación, se van a describir detalladamente los componentes que componen al sistema:

Tabla28. FrontEndJplanner

ID	1	Nombre	Front-EndJplanner	Tipo	Componente
Descripción	<p>Este componente representa todo el módulo front-end de la aplicación, el cual se encuentra desarrollado en Angular. Este componente se encuentra conformado por otros 8 subcomponentes los cuales representan componentes de angular y se encuentran diagramados bajo el estereotipo angular-component. Cada componente ejemplifica un módulo en Angular el cual es utilizado por la aplicación para pintar en pantalla las distintas funcionalidades disponibles. Adicionalmente, el componente Shared es el encargado de consumir los servicios web que son expuestos por nuestro Back-End desarrollado en Spring.</p> <p>Como se mencionó anteriormente se utiliza el patrón MVC, se puede ver aplicado en este componente ya que en Angular se ha programado teniendo una vista, un controlador para cada vista y un modelo que permite las interacciones, las vistas se encuentran en formato html y css, los controladores y modelos en formato typescript.</p>				

Tabla29. **BackEndJplanner**

ID	2	Nombre	Back-EndJplanner	Tipo	Componente
Des-crip-ción	<p>El presente componente representa el módulo back-end de Jplanner, este módulo se encuentra desarrollado en la herramienta Spring, es el encargado de proveer las interfaces para que el componente front-end consuma los servicios web que este tiene desarrollados. También, es el encargado de comunicarse con componentes externos como lo son los componentes de PeopleSoft y AuthenticationService, los cuales proveen servicios web expuestos por la DTI. Adicional a esto es el encargado de hacer el mapeo de las entidades guardadas en la base de datos del sistema, con el fin de restaurar las sesiones de los usuarios en el sistema.</p> <p>Este se encuentra compuesto por un conjunto de paquetes conformados a su vez por clases, con los cuales se encarga el sistema de proveer la información necesaria para el front-end del sistema. Como se mencionó anteriormente se mapean los datos guardados en la base de datos como entidades, para la recuperación de datos de los usuarios y se reciben los datos proveídos por los servicios web de la DTI</p>				

Tabla30. **AuthService**

ID	3	Nombre	AuthenticationService	Tipo	Componente
Des-crip-ción	<p>Este componente ejemplifica los servicios de OracleSingleSignOn sobre los cuales se encontrará funcionando la aplicación Jplanner, el usuario accede a través del sistema MAX de la Pontificia Universidad Javeriana y de esta manera obtiene acceso a la aplicación con un solo logeo.</p>				

Tabla31. PeopleSoft

ID	4	Nombre	PeopleSoft y WebServices	Tipo		Componente	
Descripción	<p>Ambos componentes hacen parte del sistema PeopleSoft de la Pontificia Universidad Javeriana, el componente PeopleSoft es el que ejemplifica al sistema de la universidad, que a través del componente WebServices provee la interfaz necesaria para el consumo de los servicios web expuestos por la DTI y que son consumidos por el back-end del sistema Jplanner. Para la descripción de cómo deben desarrollarse los servicios web se escribe un documento sobre el cual se especifican a detalle todos los elementos que debe tener cada servicio para que funcione y se integre de manera adecuada con el sistema Jplanner, para más información dirigirse a los anexos del documento SDD, anexo Servicios.docx</p>						

Tabla32. JplannerDB

ID	5	Nombre	JplannerDB	Tipo		Componente	
Descripción	<p>Este componente es la base de datos del sistema, esta base de datos se encuentra sobre el sistema PostgreSQL el cual es un sistema de gestión de datos relacional orientada a objetos. En dicha base de datos se almacena la información de cada sesión de usuario, es decir se almacenan datos como materias agregadas por alternativa, datos básicos del usuario, bloqueos realizados entre otros. Este componente se comunica con el controlador de bases de datos en el back-end del sistema.</p>						

Tabla33. Cliente

ID	6	Nombre	Cliente	Tipo		Componente	
Descripción	<p>A pesar de que dicho componente no se encuentra explícito en el diagrama, es importante mencionar que en las máquinas de los clientes ya que es utilizado Angular, el componente Front-EndJplanner será desplegado, mediante la utilización de un servidor como Apache para el despliegue web estático.</p>						

Vista física del sistema

La aplicación Jplanner se encuentra desarrollada bajo una arquitectura Cliente/Servidor, en esta sección se desarrolla una descripción detallada de todos los componentes físicos que componen e interactúan con el sistema [20]. Con el fin de apoyar esta sección se desarrolla un diagrama de despliegue el cual puede ser encontrado a continuación en la siguiente imagen:

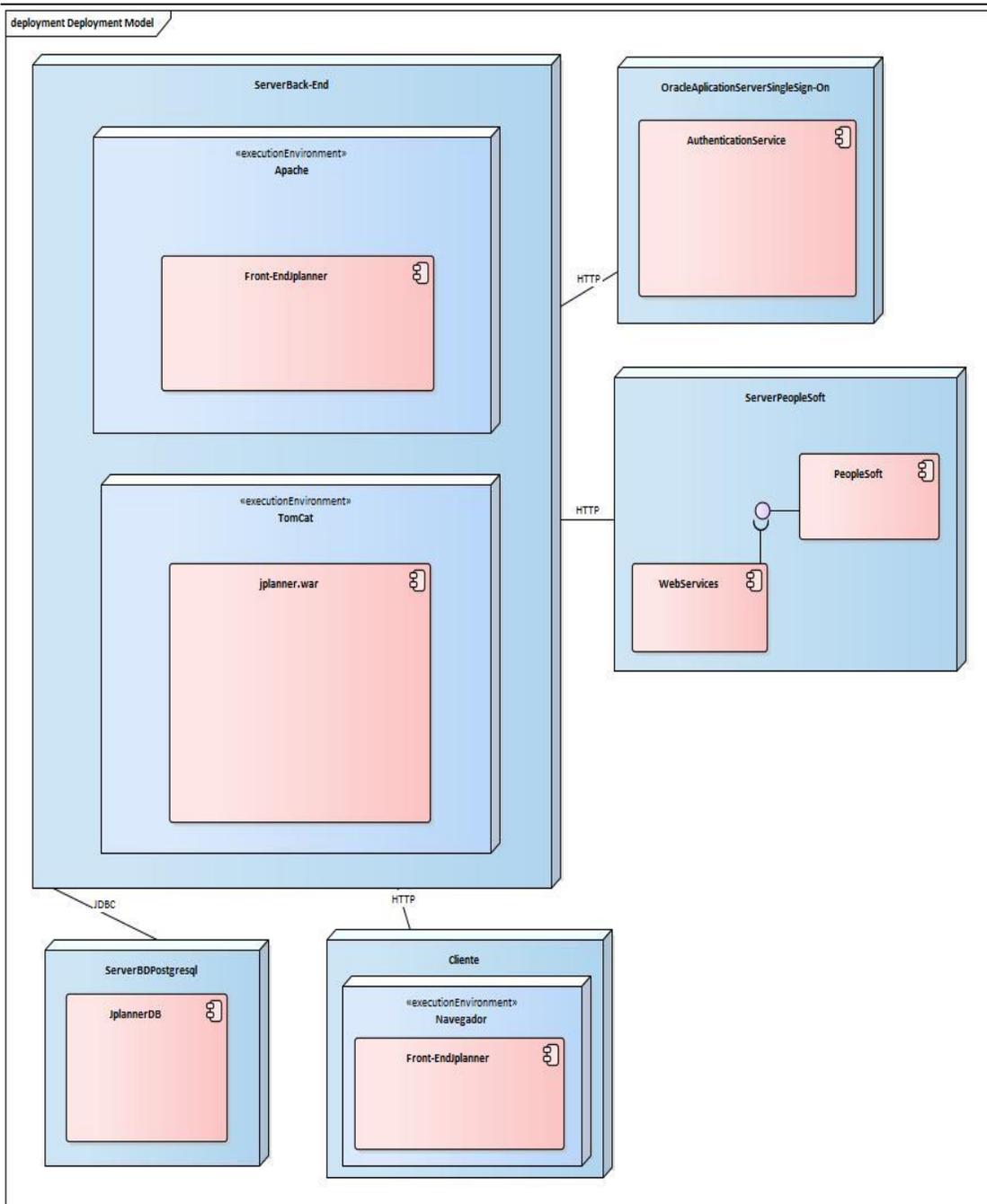


Imagen7.Diagrama Despliegue

Es importante recalcar que a pesar de poseer servicios web propios que proveen información como se ha explicado en secciones anteriores, al ser una simulación de los servicios reales expuestos por la DTI, tanto el servidor de que maneja Spring como el que será manejado por Angular, se pueden iniciar en un mismo nodo físico.

Adicionalmente, se explicarán qué nodos interactúan con nuestro sistema y como se encuentra montado el sistema de bases de datos[20].

Tabla34.Descripción Nodos

ID	1	Nombre	ServerBack-End	Tipo	Nodo
Des- crip- ción	Este nodo representa el servidor sobre el cual está montado todo el sistema es decir tanto el componente Front-End como el Back-End, este servidor es propiedad de la DTI.				
Des- crip- ción	1.1	Apache		Execution Environment	
	Este nodo representa el ambiente de ejecución sobre el cual se va a correr el componente Angular, este es el servidor web ya que sobre este se corren las páginas de la aplicación.				
Des- crip- ción	1.2	TomCat		Execution Environment	

	<p>Este nodo representa el ambiente de ejecución sobre el cual se montará el componente Spring, este servidor se enfoca en ser un contenedor de servlets y JavaServerPages, razón por la cual fue seleccionado. En este nodo se encuentra contenido todo el Back-End del sistema, los métodos de conexión con la base de datos y los componentes de conexión y exposición de servicios web.</p>		
Des-crip-ción	2	ServerBDPostgreSQL	Nodo
	<p>Nodo usado para representar la base de datos relacional de Jplanner, este nodo se encuentra desarrollado bajo el sistema PostgreSQL utilizado por ser un sistema orientado a objetos. Este nodo posee la base de datos JplannerBD como subcomponente, la conexión con Jplanner se realiza a través de JDBC..</p>		
Des-crip-ción	3	OracleAplicationServerSingleSingOn	Nodo
	<p>Este nodo es usado para representar el sistema de autenticación propuesto por la DTI el cual utiliza el método SingleSignOn de la empresa Oracle. Este nodo se conectará al sistema a través de HTTP.</p>		

Des- crip- ción	4	ServerPeopleSoft	Nodo
	Este nodo representa el servidor sobre el cual está montado el sistema PeopleSoft de la Universidad, sistema desde el cual se proveerán servicios web para el consumo de información necesaria para el adecuado funcionamiento del sistema, posee dos componentes los cuales representan al sistema PeopleSoft y el componente encargado de exponer servicios web. La conexión con Jplanner se realiza mediante HTTP.		
Des- crip- ción	5	Cliente	Nodo
	En este nodo se hace referencia al cliente el cual podrá utilizar el sistema bien sea en un celular, tablet o pc. El cliente se conecta al sistema mediante el protocolo HTTP.		
Des- crip- ción	5.1	Navegador	Execution Environment
	Este es el ambiente de ejecución sobre el cual el cliente lanza la aplicación, sobre este se encontrará el componente Front-End del sistema ya que se usa la tecnología Angular.		

--	--

Vista de procesos del sistema

En esta sección se presentarán las principales acciones o procesos que un usuario puede realizar en el sistema, brindando un enfoque más cercano a los casos de uso, pero guardando la relación con los diagramas de navegabilidad presentados en la sección 7.3 del presente documento. El objetivo de esta sección es brindar a los desarrolladores una abstracción de alto nivel de los principales procesos que debe ser capaz de ejecutar el sistema y el usuario. Con el fin de facilitar la comprensión de los procesos se ha desarrollado un diagrama BPMN el cual ejemplifica el flujo de acciones que realiza el sistema.

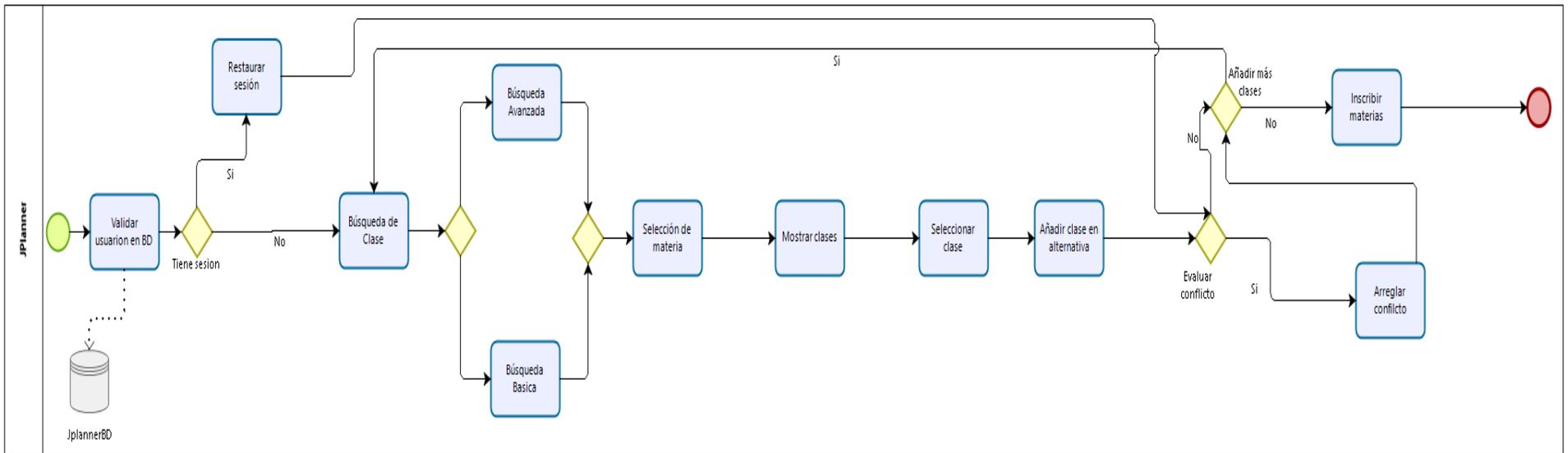


Imagen8.Diagrama acciones

En el anterior diagrama se muestran las principales operaciones del sistema como lo es recuperar una sesión de la base de datos, permitir realizar una búsqueda, mostrar las materias resultado, seleccionar una clase lo cual hace referencia a lo que es entendido en otros contextos como una sección, cada materia se compone de muchas clases o secciones.

Agregar materias en una alternativa en este caso es importante aclarar que una alternativa ejemplifica una opción de horario de las 6 diferentes que permite el sistema. También, es posible arreglar un conflicto el cual se genera cuando dos o más clases se cruzan en un mismo horario, por último, se puede terminar el proceso inscribiendo materias lo que invoca un servicio web expuesto por la DTI.

Estructura del sistema

En esta sección se hablará acerca de la organización de cada componente de la aplicación JPlanner, esto será explicado mediante diagramas de clases que representan una manera gráfica y de manera estática la estructura general de un sistema[18]. Por otro lado, es necesario explicar que se describirán componentes tanto del back-end(Spring) como del front-end(Angular), a continuación, se hablará del back-end.

Primero que todo, el back-end de la aplicación JPlanner está compuesto de siete paquetes que a su vez tienen por dentro clases, cabe resaltar que se usó el modelo-vista-controlador, sin embargo, no existen vistas en ningún componente de Spring debido a que es el back-end de la aplicación, por lo tanto, la mayoría de clases representan entidades y controladores tanto de estas dichas entidades y de los servicios web expuestos por la DTI. A continuación, se mostrará el diagrama completo para así explicar cada paquete que conforma el mismo.

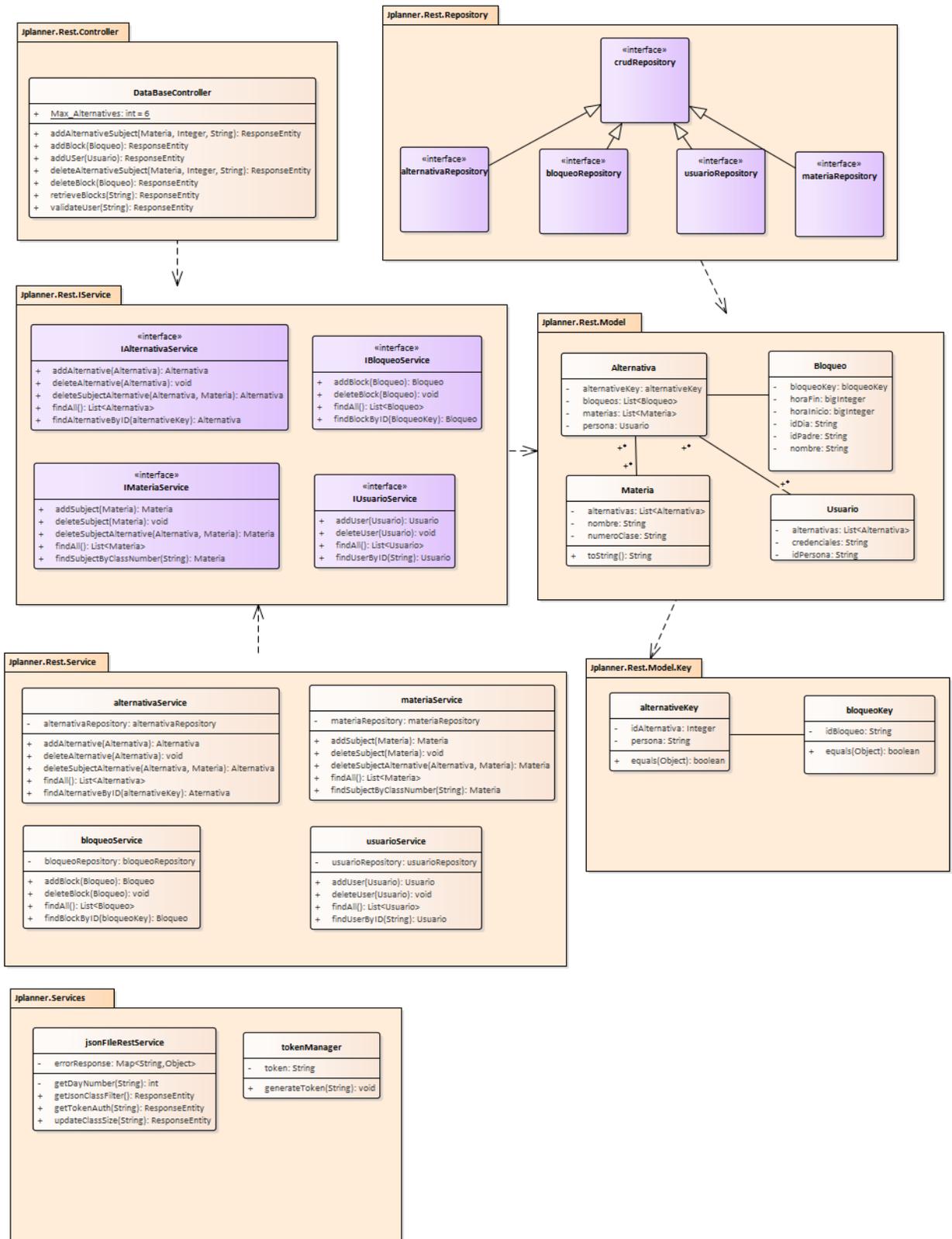


Imagen9.DiagramaDeClasesSpring

Como se puede ver en la imagen 5 hay 7 paquetes, en donde cada uno de ellos, contienen clases encargadas de una determinada función, a continuación, se hará una breve explicación de cada dicha función:

- Jplanner.Rest.controller: Este paquete tiene solamente una clase llamada DataBaseController que se encarga de proveer servicios web a nuestro front-end para restaurar la sesión de un determinado usuario ya autenticado en el sistema.
- Jplanner.Rest.Iservice: Este paquete tiene 4 interfaces que se encargan de definir los métodos para que otras clases los implementen, todos estos métodos sirven para obtener información de la base de datos.
- Jplanner.Rest.Service: Este paquete tiene 4 clases que se encargan cada una de obtener información de la base de datos con respecto a una entidad, estas clases implementan todos los métodos de su respectiva interfaz.
- Jplanner.Rest.Model: Este paquete tiene 4 clases, las cuales se encargan de mapear y representar las entidades de la base de datos y sus relaciones.
- Jplanner.Rest.Model.Key: Este paquete tiene 2 clases y sencillamente sirven para representar las llaves primarias compuestas de las entidades Alternativa y Bloqueo.
- Jplanner.Rest.Repository: Este paquete tiene 1 interfaz y otras 4 que heredan de ella, estas interfaces se encargan de definir los métodos (CRUD) específicos de una entidad para que así otras clases los implementen.
- Jplanner.Services: Este paquete tiene solamente dos clases muy importantes para el funcionamiento de la aplicación, la primera llamada jsonFileRestService se encarga de exponer servicios a nuestro front-end y la segunda clase llamada tokenManager se encarga de consumir el servicio web de la DTI que retorna un token de autorización que los demás servicios requieren.

Para continuar, el front-end de la aplicación Jplanner se encarga de mostrar toda la información obtenida de los diferentes servicios web expuestos por la DTI, cabe resaltar que desde el back-end se llaman estos servicios, sin embargo, desde nuestro front-end se consumen los servicios expuestos por nuestro back-end y dichos servicios se encargan de consumir los de la DTI. A continuación, se mostrará el diagrama de clases y se explicará.

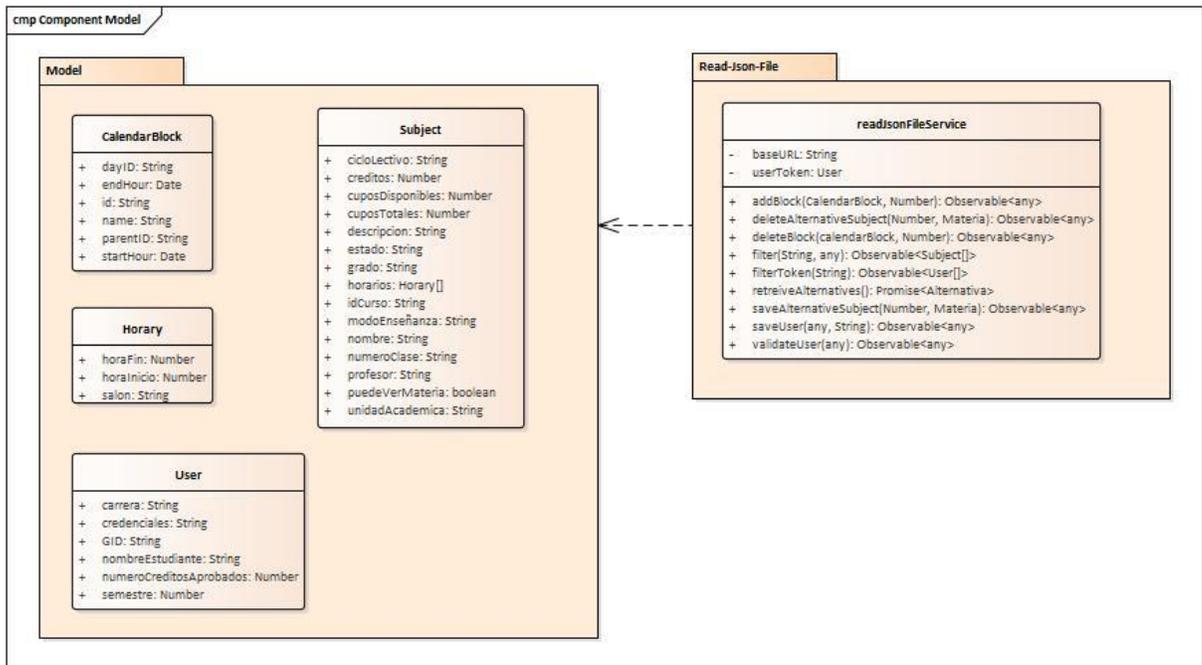


Imagen10.DiagramaDeClasesAngular

Como se puede ver en la imagen 6 solo hay 2 paquetes, en donde cada uno de ellos, contienen clases encargadas de una determinada función, a continuación, se hará una breve explicación de cada dicha función:

- **Model:** Este paquete representa al modelo del MVC, todas las clases se encargan de representar entidades de la base de datos y de los resultados de los servicios web ya consumidos por la DTI.
- **Read-Json-File:** Este paquete solamente tiene una clase llamada readJsonFileServices que se encarga de consumir todos los servicios expuestos por nuestro back-end.

Para concluir, es necesario resaltar que tanto en el front-end como en el back-end se utilizaron el modelo-vista-controlador teniendo en cuenta que para el back-end no hubieron vistas debido a que el sistema desarrollado en Spring sencillamente se encarga de la lógica de negocio y de integración con el sistema de la dirección tecnológica de información de la Pontificia Universidad Javeriana.

Los siguientes atributos de calidad de nuestro software fueron desarrollados en nuestro back-end como, por ejemplo: alta recuperación, interoperabilidad y tolerancia a fallos. Otro aspecto importante de resaltar es que dicho back-end se encarga de crear servicios para que el front-end los consuma y muestre los resultados y así de esta manera conjuntamente puedan hacer que Jplanner funcione en su normalidad.

Persistencia

Un diagrama de entidad relación representa la manera en que objetos, conceptos o personas son representadas en un sistema de software. Por otro lado, es una herramienta para el modelado de datos, para así representar entidades relevantes, sus interrelaciones y propiedades y de esta manera se pueda conocer la manera en que los datos de un determinado sistema son almacenados en forma persistente. Esta sección comprende el diagrama anteriormente mencionado junto con la descripción de cada entidad y sus relaciones.

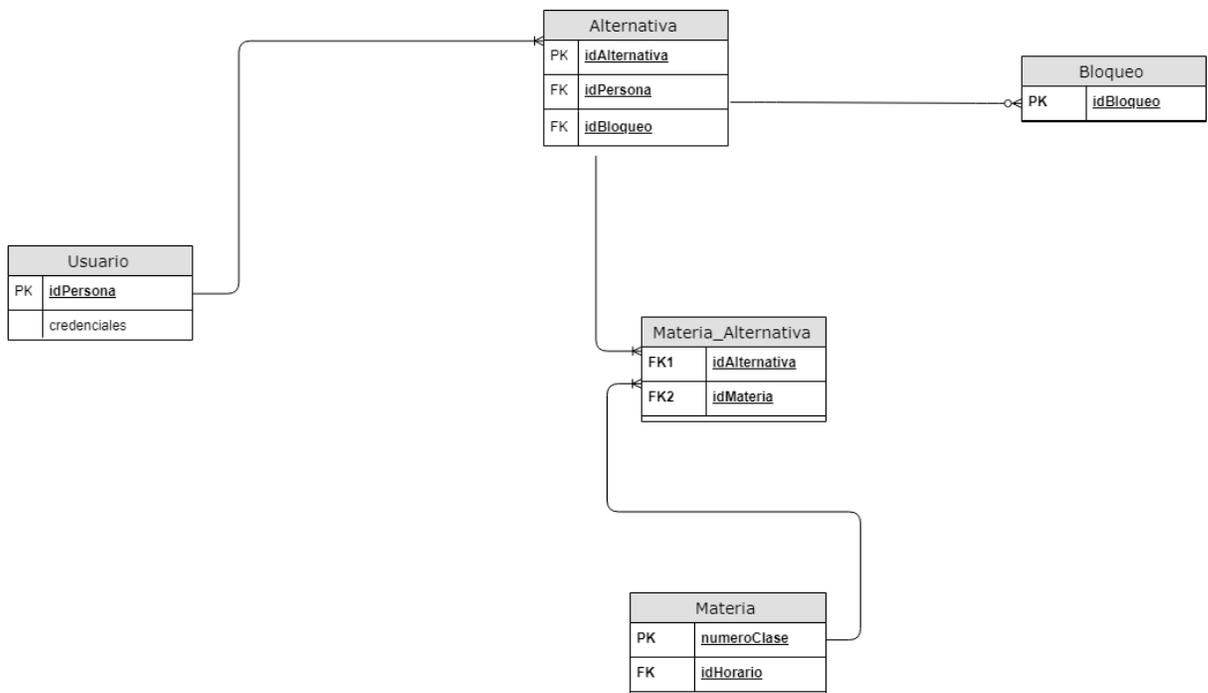


Imagen11.Modelo De Datos BD

Para comenzar, es necesario explicar por qué se realizó un modelo de datos tan sencillo y pequeño teniendo en cuenta la cantidad de datos que se utilizan en JPlanner. Como se sabe la mayoría de información mostrada al usuario por medio de nuestra interfaz proviene de servicios web tipo Rest expuestos por la dirección de tecnología de información de la Pontificia Universidad Javeriana y consumidos por nuestro componente JPlanner, sin embargo, un requisito mencionado por el director de admisiones y registro fue que nuestro aplicativo web debía guardar la sesión de todos los usuarios que ingresaran y así mismo todo lo que hicieron dentro de la misma, es decir, si un estudiante ya autenticado ingresa a JPlanner y construye varias alternativas de horario que contengan materias y bloqueos y por alguna razón el navegador se cierre al volver a ingresar al aplicativo web este dicho usuario ya autenticado podrá visualizar en las alternativas de horario todo lo que tenía ya planeado sin perder progreso alguno, garantizando así la fiabilidad del sistema la cual es un atributo de calidad muy

importante a la hora de hablar de la capacidad de un software para recuperarse ante fallos.

Cabe resaltar que se utilizaron solamente 2 tipos de relaciones en bases de datos los cuales son:

- **Uno a muchos:** Como se ve en la imagen número 6 la relación entre Usuario y Alternativa y entre Alternativa y bloqueo es uno a muchos lo cual indica que un usuario puede tener varias alternativas y una alternativa puede tener varios bloqueos; Las razones de esto serán explicadas más adelante.
- **Muchos a muchos:** Como se ve en la imagen número 6 la entidad Materia_Alternativa representa una entidad de unión entre Alternativa y Materia ya que las alternativas pueden tener varias materias, y así mismo las materias varias alternativas; La razón de esto será explicada más adelante.

Las entidades de la imagen 7 representan todos los objetos necesarios del sistema para poder restaurar una sesión las cuales son:

- **Usuario:** Representa a cada persona que use el sistema y ya se haya autenticado,
- **Alternativa:** Representa todas las opciones de horario que el usuario haya creado y planeado.
- **Bloqueo:** Representa a todos los bloqueos que se creen dentro de una tentativa de horario.
- **Materia:** Representa todos los cursos obtenidos de los servicios web de la DTI.
- **Materia_Alternativa:** Es una tabla intermedia entre dos entidades que tienen relación muchos a muchos las cuales son materia y alternativa.

Es necesario resaltar que es muy importante la persistencia en nuestro sistema ya que es la manera de asegurar una alta recuperación ante cualquier inconveniente externo e interno, siempre que el usuario ya se haya autenticado y sea redirigido a JPlanner se persistirán todas las alternativas que este mismo realice, en donde dentro de ellas hay materias y bloqueos que serán inscritos en la formalización de matrícula del correspondiente ciclo lectivo.

Interfaz de usuario

La aplicación de JPlanner tiene una interfaz intuitiva y amigable al usuario debido a que uno de los requisitos no funcionales en cuanto a atributos de calidad se refiere fue la usabilidad, las pantallas se diseñaron de tal manera que una persona pueda usar el sistema sin necesidad de haber tomado una capacitación al respecto, También, es importante mencionar que se llevó a cabo el desarrollo de dos alternativas de mockups las cuales son las que definen los diseños finales de la aplicación, estos mockups fueron sometidos a pruebas con un grupo conformado por la Universidad los cuales evaluaron las soluciones, tomaron decisiones sobre cuál de las propuestas

fue la mejor y que ajustes podrían ser realizados a la solución final. Para obtener más información acerca de los mockups a detalle, diríjase a los anexos del documento SDD, nombre anexo Mockups con manual de identidad 3.pdf.

Por otro lado también la portabilidad fue tomada en cuenta a la hora de desarrollar JPlanner, para que así cualquier estudiante pueda inscribir materias durante su cita desde un celular, una Tablet y/o un computador de escritorio. A continuación, se presentan los diagramas de navegabilidad de dichos dispositivos [19]:

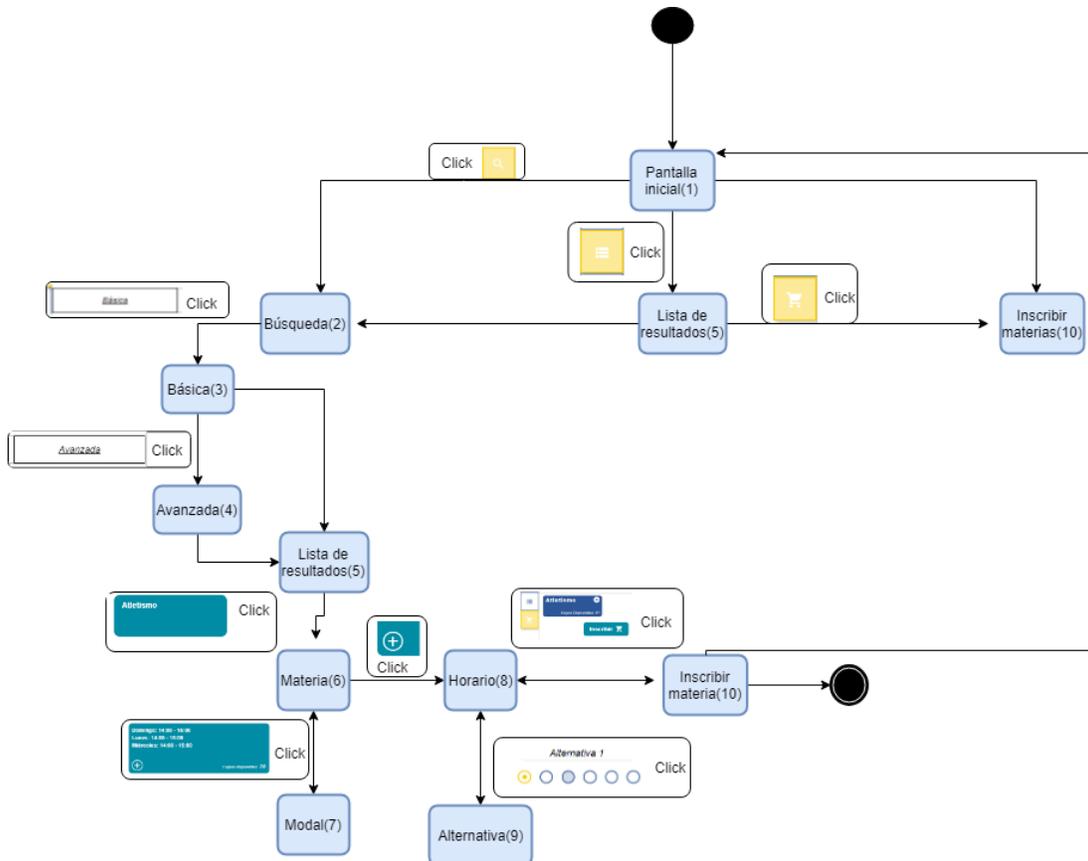


Imagen12.Diagrama navegabilidad PC y Tablet

La imagen 8 muestra el flujo de pantallas de la aplicación JPlanner en computador de escritorio y en tablet, primero que todo es necesario resaltar que para navegar entre los diferentes estados del diagrama se debe realizar click en los botones y funcionalidades del sistema que son mostrados en la imagen, por el otro lado, las asociaciones que no tienen texto alguno, es decir click, significa que el sistema lleva al usuario automáticamente a esta misma sin necesidad de intervención alguna, cabe resaltar que la única diferencia entre la interfaz de usuario de computador de escritorio con la de tablet es la ubicación del calendario en la pantalla como se ve a continuación:

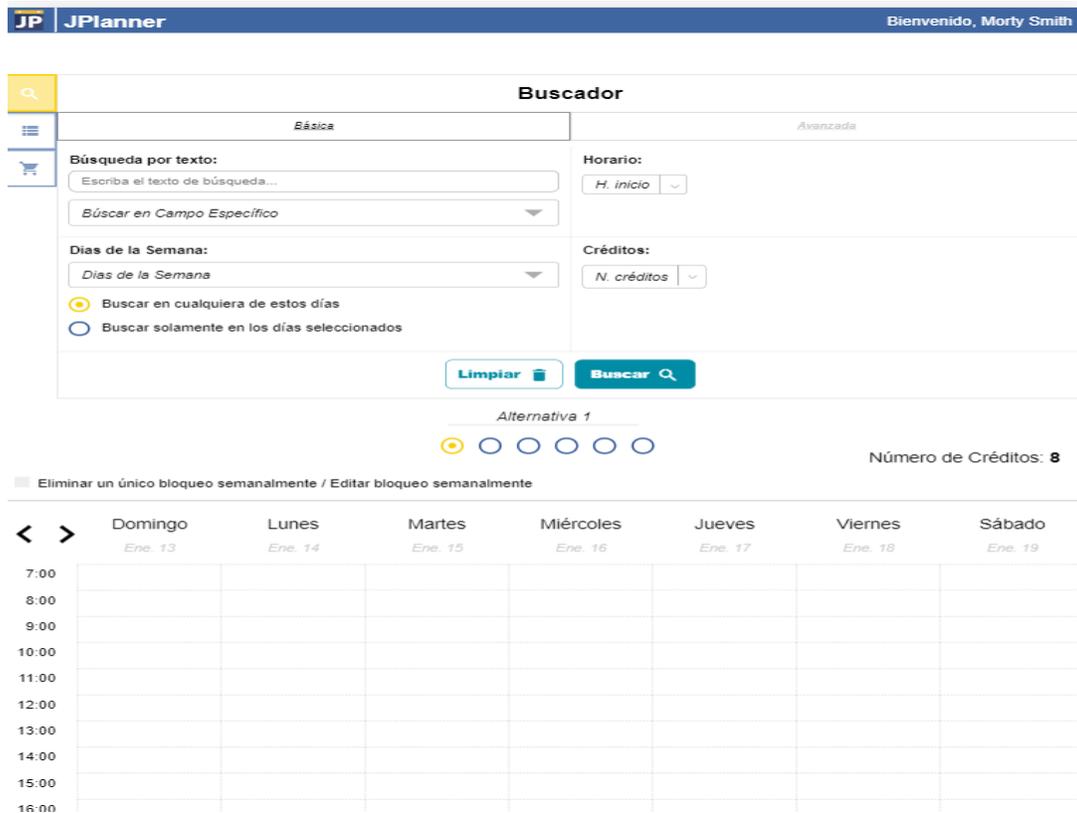


Imagen13.Ubicación calendario tablet

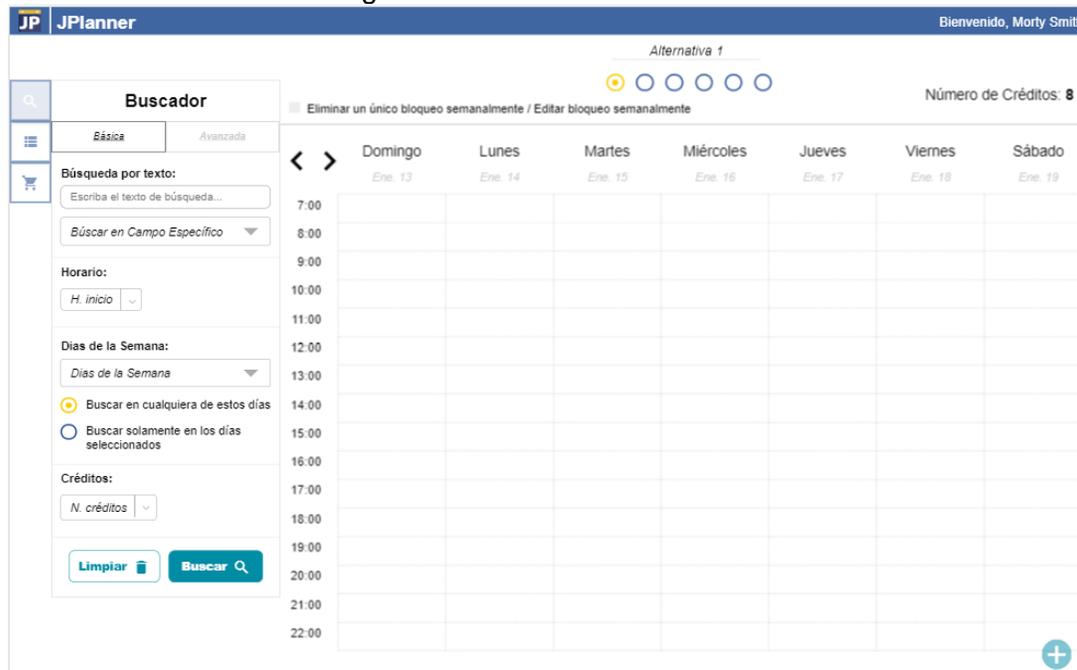


Imagen14.Ubicación calendario pc

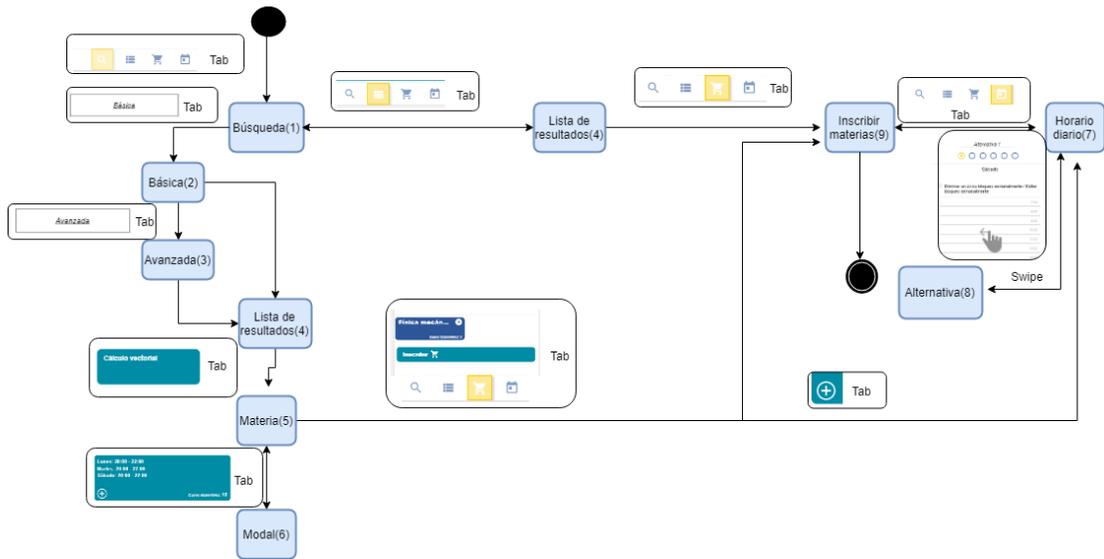


Imagen15.Diagrama navegabilidad mobile

En la imagen 11 se ilustra el flujo de pantallas en mobile[19], la interfaz de usuario en este dicho dispositivo si cambia bastante debido a que ahora la pantalla inicial es la pestaña de búsqueda y además se añade una barra en la parte inferior de la pantalla para navegar entre pestañas más fácil e intuitivamente, también el calendario cambia debido a la restricción del tamaño de pantalla, es decir, la visualización del calendario no es semanal sino diario. Es importante aclarar que al utilizar la aplicación JPlanner en un dispositivo móvil, automáticamente el sistema activará todos los gestos que una pantalla táctil debe tener para hacer aún más intuitivo el uso de la aplicación, a continuación, se describirán los componentes de los diagramas de navegabilidad:

- **Búsqueda:** Componente que representa toda búsqueda que se realice para filtrar materias, este componente tiene todos los campos para personalizar mejor todos los resultados de una determinada búsqueda realizada por un usuario, estos campos corresponden a búsquedas básicas y avanzadas.
- **Básica:** Componente que representa a la búsqueda básica.
- **Avanzada:** Componente que representa a la búsqueda avanzada.
- **Lista de resultados:** Componente que muestra la lista de resultados que cumplan con los campos diligenciados tanto en la búsqueda básica como en la avanzada.
- **Materia:** Componente que representa todas las materias mostradas en la lista de resultados, este componente solo muestra el nombre de la materia.
- **Horario:** Componente que representa todas las clases de una materia, muestra el horario en el cual se brinda dicha clase junto con la cantidad de cupos disponibles de la misma.
- **Modal:** Componente que muestra la información de una materia.

- **Alternativa:** Componente que representa el calendario semanal en el sistema, en este componente se ven reflejadas las clases añadidas y los bloques creados
- **Inscribir materia:** Componente que muestra las materias a inscribir en la formalización de matrícula

Para concluir, se puede decir que se describieron los diagramas de navegabilidad de los 3 dispositivos que soporta la aplicación JPlanner junto con los componentes que los conforman, así mismo se demuestra la manera en que la portabilidad y la usabilidad del sistema fueron abordadas para que así la experiencia de usuario sea muy buena generando una alta satisfacción a los estudiantes a la hora de inscribir materias.

VI- DESARROLLO DE LA SOLUCIÓN

En esta sección se explicará todo el proceso de desarrollo de la solución desde su fase inicial hasta la actual, también se mostrará el resultado final de la aplicación JPlanner y junto a esto una breve descripción de todas las funcionalidades. Primero que todo es necesario decir que se llevaron a cabo 3 fases las cuales son:

Pre-desarrollo

En esta fase se realizó el levantamiento de los requerimientos que surgen a raíz del problema y los que adicionalmente propone la DTI. También se realizaron modelos interactivos que permitan dar una idea del prototipo a implementar. Por último, con compañía de la DTI y del director de trabajo de grado, se establecerá el alcance del trabajo de grado.

- **Método:** Como método en esta primera etapa se utilizará una adaptación de Scrum donde se realiza una priorización de los requerimientos, con esto se puede esclarecer cuales requerimientos son los más importantes a cumplir con el prototipo. Adicionalmente se utilizan historias de usuario donde se realiza un manejo mucho más adecuado de lo deseado por parte de la DTI como usuario. Es de vital importancia que esta etapa se cumpla debido a que gracias a estos requerimientos por parte de la DTI definen el alcance final que tendrá el trabajo de grado y a que debe comprometerse el equipo de trabajo.
- **Actividades:** Para esta fase se proponen las siguientes actividades:
 - Levantamiento de requerimientos
 - Desarrollo de modelos interactivos para el usuario
 - Definición del alcance
 - Permisos de uso de la información
 - Definir Stakeholders y responsables de la información.

- **Resultados esperados:** Se espera tener el documento formal de los requerimientos finales pactados con la DTI, el alcance del proyecto de grado pactado entre el equipo y la DTI y el modelo interactivo seleccionado por la DTI como el indicado para basarse en el desarrollo posterior.

Desarrollo

En esta fase se realizó el diseño formal de la aplicación y el desarrollo total del prototipo, utilizando pruebas en conjunto con la DTI y así evaluar los avances en cuanto al código, y reuniones formales con los expertos en diseño para que sean aprobados los diseños respectivos de la aplicación(mock-ups).

- **Método:** Como metodología utilizada en esta fase se adaptará XP y SCRUM. Scrum es una metodología ágil que permite el trabajo en equipo, organizado y que tiene en cuenta en todo el proceso a los Stakeholders del proyecto. Por otra parte, Xp es una metodología centrada en el código, lo cual es importante debido a que se pretende realizar una documentación muy profunda de todo el código desarrollado, adicionalmente, el uso de programación en parejas que permita minimizar la curva de aprendizaje en nuevas herramientas.
- **Actividades:** Para esta fase las actividades propuestas se toman en base a el diseño y el desarrollo del prototipo y son las siguientes:
 - Diseño de diagramas
 - Diseño de arquitectura de aplicación
 - Muestra de diagramas a la DTI
 - Retroalimentación por parte de la DTI
 - Delegación de tareas a los integrantes del equipo
 - Sprints de inicio y fin
 - Revisiones cruzadas en reuniones intermedias
 - Revisiones junto con la DTI
 - Desarrollo de pruebas
 - Entrega de prototipo
- **Resultados esperados:** Se espera la entrega de todos los diseños necesarios para el desarrollo del prototipo, junto con el prototipo probado por la DTI y aprobado.

Documentación

En la última fase no se utiliza una metodología en específico debido a que se consolida toda la documentación que se desarrolló a lo largo del trabajo de grado y se

deja abierta la posibilidad de un posible trabajo futuro. Por lo que en esta etapa se utilizan estándares de distintos documentos como el SRS, el SPMP y el SDD.

- **Método:** Como método se toman como base los estándares utilizados en el SPMP, el SRS y el SDD como es el ISO/IEC/IEEE 1016, ISO/IEC/IEEE 42010, y el modelo 4+1.
- **Actividades:** Las actividades planeadas para esta etapa son:
 - Consolidación del SRS
 - Consolidación del SPMP
 - Consolidación del documento final del trabajo de grado
- **Resultados esperados:** Como resultados esperados de esta etapa se tienen los documentos respectivos antes mencionados, agregando que estos deben estar aprobados por el director de trabajo de grado.

Es necesario explicar las metodologías que se usaron para todo el desarrollo del proyecto las cuales fueron Scrum y XP, estas dichas metodologías fueron escogidas debido a todas las ventajas que proveen, cabe resaltar que no se siguieron al pie de la letra, sino que se escogieron cosas de cada una para así formar un “híbrido” entre las dos, a continuación, se mostrará mediante una tabla la cantidad de horas-hombre trabajadas en las fases anteriormente descritas:

	Sp-1	Sp-2	Sp-3	Sp-4	Sp-5	Sp-6	Sp-7	Sp-8	Sp-9	Sp-10
Pre-desarrollo	105									
Desarrollo		105	57	216	183	222	198	261	87	51
Documentación									50	50

Tabla35. Horas-hombre trabajadas

Como se puede observar en la tabla 8 se trabajaron 1585 horas-hombre durante todas las fases del proyecto, esto se calculó con ayuda de la metodología scrum, es decir, en cada sprint se asignaban historias de usuario a cada integrante del grupo y a estas se les consideraban una determinada cantidad de story-points lo cual indica la dificultad de la tarea a realizar y el tiempo que toma realizarla, por otro lado, se definió en la fase de Pre-desarrollo que cada story point serian 3 horas-hombre de trabajo. Por otro lado, también se trabajaron 44 horas-hombre en reuniones del equipo y con la DTI, para ver esto mejor, en los anexos están dichas actas de reuniones. Es importante aclarar la manera en la que se llevaron a cabo todas las planeaciones de sprints y la asignación de historias de usuario a cada integrante, la cual es mediante el uso de la herramienta trello que es para la administración de proyectos de software, en donde hay un dashboard donde fueron creados todos los 10 sprints y en cada uno

historias de usuario que eran asignadas a un determinado integrante del grupo junto con su nivel de dificultad(storypoints), a continuación se presenta una breve imagen para mostrar el uso de dicha herramienta:

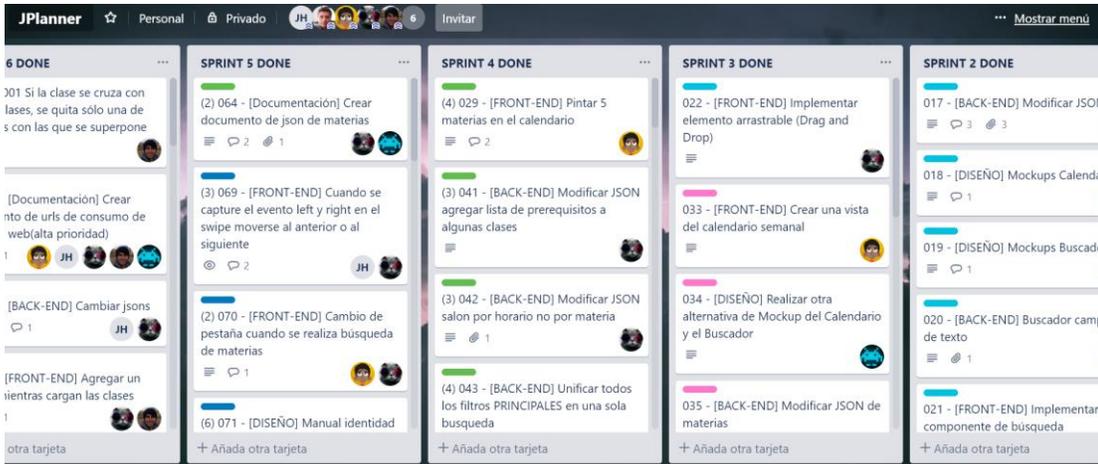


Imagen16.Trello

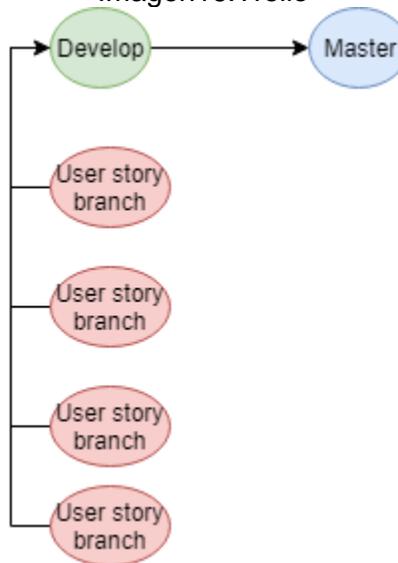


Imagen17.Github

Para continuar, como se puede ver en la imagen 13, con la herramienta GitHub fue la manera en que se manejó las versiones de cada implementación del proyecto, cada nueva historia de usuario era una nueva rama que tenía que ser probada y aprobada por un determinado integrante del grupo y así de esta manera se unía automáticamente con la rama principal llamada develop en donde quedaron todas las funcionalidades desarrolladas, esto fue de gran ayuda debido a que a medida que se iban implementando requisitos del sistema, estos también eran probados antes de unirlos con las demás funcionalidades lo cual ocasiono que existiera un proceso de validación

de calidad antes de establecer una historia de usuario como cerrada y ya implementada.

Para concluir, se mostrará la solución implementada mediante imágenes para que así el lector pueda ver el resultado final de todo el proceso de desarrollo del proyecto JPlanner anteriormente descrito:

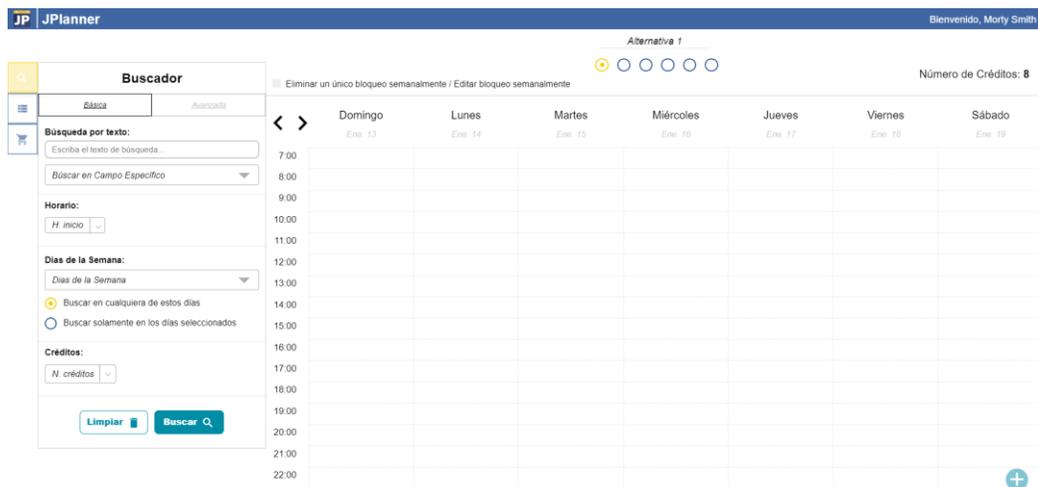


Imagen18.Solución(1)

En esta imagen se ve la pantalla inicial de JPlanner, a la izquierda se puede ver el buscador y el panel para cambiar a lista de resultados o inscribir materias y a la derecha se puede ver el calendario junto con las 6 alternativas y el número de créditos de cada alternativa.

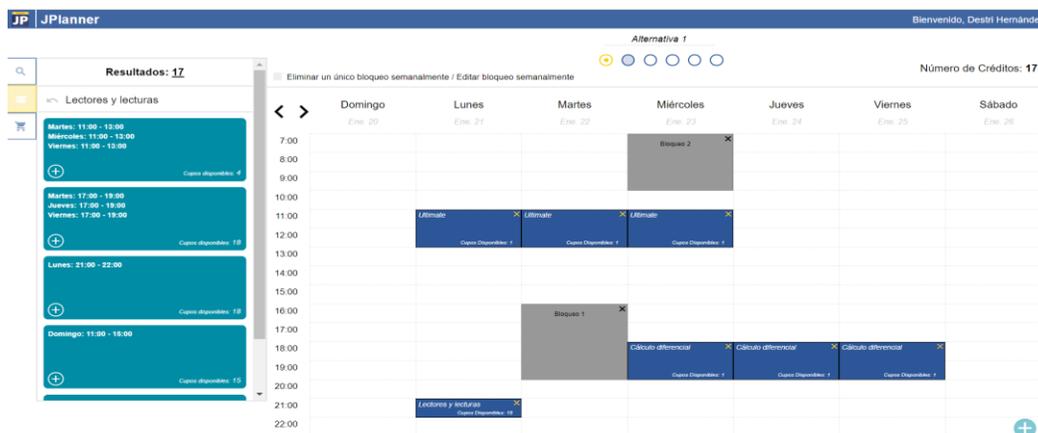


Imagen19.Solución(2)

En esta imagen se ve la lista de resultados que cumplen con los filtros de la búsqueda, se pueden ver bloqueos creados dentro del calendario y así mismo materias añadidas en la alternativa 1.

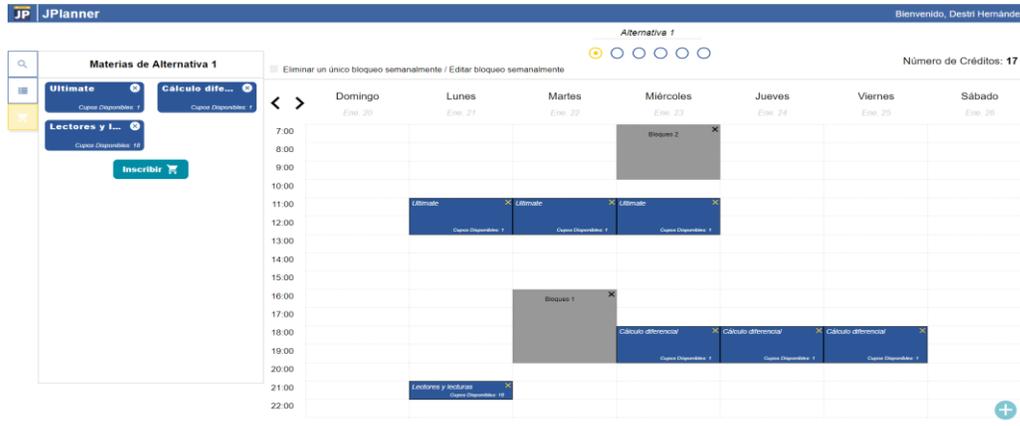


Imagen20.Solución(3)

En esta imagen se pueden ver las materias que van a ser inscritas de la alternativa 1 y así mismo el calendario a ser formalizado en la matricula del estudiante autenticado como se puede ver en la esquina superior derecha.

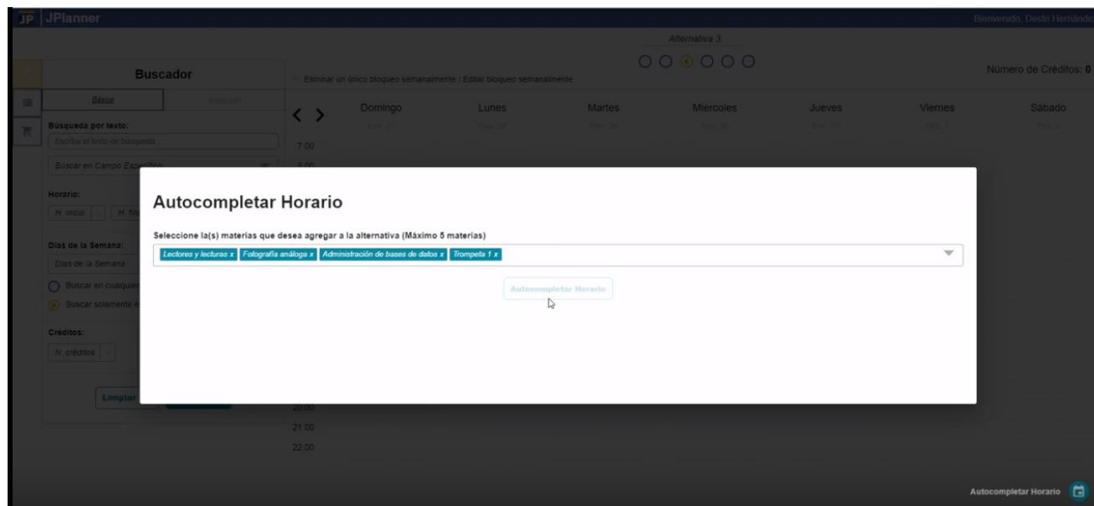


Imagen21.Solución(4)

En esta imagen se puede ver una de las funcionalidades más importantes del sistema JPlanner, la cual es, la recomendación de materias mediante un algoritmo.

Para concluir el apartado, se mostraron pantallazos de la solución implementada en donde se pueden apreciar ciertas funcionalidades del sistema como lo son: la búsqueda de materias, la lista de resultados de un determinado filtro, la inscripción de las materias dentro de una alternativa y por último el algoritmo de recomendación de materias, cabe resaltar que no todas las funcionalidades se pueden ver en dichas imágenes ya que se vuelve complicado explicarlas mediante pantallazos debido a que el sistema está desarrollado para computadores de escritorio, tablets y celulares.

VII- RESULTADOS

En este apartado se van a presentar los resultados obtenidos de las pruebas de usabilidad desarrolladas al sistema, es también importante aclarar que se desarrollaron pruebas de calidad al código realizado, las cuales permitían asegurar el correcto funcionamiento del sistema a medida que este era desarrollado. Se desarrollaron tanto pruebas de caja negra como de caja blanca, para las pruebas de caja negra se creó un formato que permite evaluar cada módulo que fuese entregado del sistema, este formato y su documentación puede ser encontrado en el documento anexo Pruebas del Sistema. Para las pruebas de caja blanca se realizó code review de cada subrama en la herramienta GitHub lo cual permitía tener un control de calidad más a detalle del código que iba siendo entregado.

Para las pruebas de usabilidad se desarrolló un formulario utilizando la metodología QUIS (Questionnaire for User Interface Satisfaction) el cual busca evaluar la experiencia y la utilidad en la herramienta, para este caso se busca una comparación entre el método de inscripción de materias convencional es decir a través de PeopleSoft y JPlanner. De esta manera se obtienen resultados subjetivos que permiten analizar un porcentaje de percepción de la aplicación. Estas pruebas fueron realizadas a estudiantes de la Pontificia Universidad Javeriana Bogotá, de distintas carreras y sin ningún acercamiento anterior al sistema lo que permite no tengan ningún sesgo o preferencia especial por Jplanner.

El formato utilizado para las pruebas de usabilidad puede ser consultado en el documento anexo PruebasDeUsabilidad. Para obtener más información a detalle acerca de la documentación de las pruebas de usabilidad por favor diríjase al documento anexo Pruebas del Sistema.

Se permitió que los usuarios tuvieran contacto con la aplicación desde distintos dispositivos como smartphones o computadores lo que de manera implícita permitía la realización de pruebas de portabilidad, ya que se probaron todos los tipos de dispositivos que el sistema debía aceptar, cada usuario fue guiado mostrándole en donde podía encontrar la guía de usuario, posterior a esto cada usuario procede libremente a usar el sistema y, por último, cada estudiante llenó el formato del cual se habló con anterioridad. A continuación, con un gráfico se profundizará y se darán las conclusiones respectivas.

Se evaluaron 23 personas las cuales permitieron responder a un conjunto de aspectos planteados en el formulario como lo son:

- Recomendación de la aplicación
- Factores generales del uso de la aplicación
- Factores visuales de la aplicación
- Terminología e información en el sistema
- Aprendizaje del sistema
- Capacidades del sistema

Teniendo cada una de estas divisiones, se realiza un conjunto de preguntas por división y adicionalmente, se permite listar aspectos positivos y negativos del sistema lo que permite tener una visión en general de que tan aceptado fue el sistema y de cuáles son las oportunidades de mejora que se presentan.

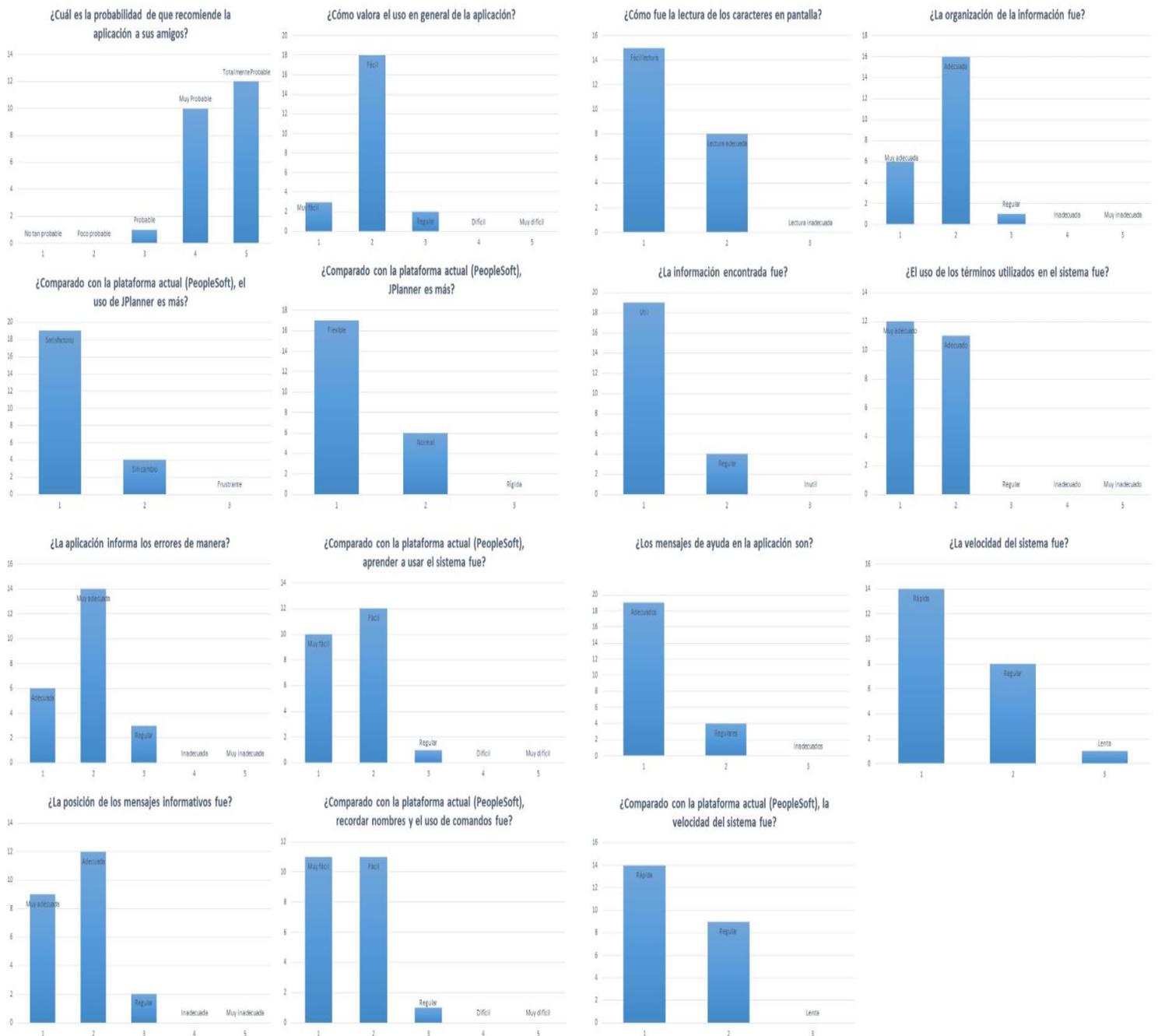


Imagen22. Pruebas Usabilidad

Tras terminar el análisis de las pruebas de usabilidad desarrollados el equipo Jplanner puede concluir que el sistema fue aceptado por todos los usuarios que fueron encuestados, como criterio para dar dicha afirmación. Se tuvieron en cuenta las franjas sobre las cuales se calificaron las distintas secciones de las que se habló anteriormente. Ninguna respuesta en cuanto a la mayoría de las secciones tuvo un valor por debajo de los términos medios de la franja de calificación, es decir no tuvo calificaciones negativas la aplicación.

Todos los usuarios concuerdan en que recomendarían a sus amigos el sistema, lo que permite demostrar la aseveración realizada anteriormente. Se puede concluir adicionalmente, que el sistema cumplió su propósito ya que en el momento de comparar Jplanner contra el sistema actual de inscripción PeopleSoft, los usuarios con sus calificaciones afirman que el sistema mejora muchos de los problemas que presenta actualmente PeopleSoft.

VIII- CONCLUSIONES

1. Análisis de impacto del proyecto

JPlanner es una aplicación hecha por estudiantes de la Pontificia Universidad Javeriana, quienes han hecho uso del sistema actual de inscripción de materias en múltiples ocasiones y conocen sus ventajas y desventajas. Como se mencionaba en la sección **1.1.2 Formulación del Problema**, existen algunos problemas respecto a la facilidad de uso del sistema y a la consulta de materias dependiendo del estudiante, es decir, que no existe una funcionalidad que permita realizar búsquedas personalizadas que se base en materias cursadas por el estudiante y que sean parte del pensum de su carrera actual.

Estos problemas necesitan ser solucionados para mejorar la usabilidad del sistema y de esta forma facilitar el momento de inscripción de materias para los estudiantes de la Universidad, realizando recomendaciones de materias a cursar y haciendo uso de un sistema gráfico.

Teniendo en cuenta estos problemas presentados por los mismos estudiantes, quienes son los usuarios principales del sistema, se llevaría a cabo la solución JPlanner como un aplicativo web que pueda ser utilizado desde diferentes dispo-

sitivos. A corto plazo se podría notar que los usuarios actuales del sistema tendrían un nivel de aceptación alto, ya que llevaría a cabo funcionalidades del sistema PeopleSoft que estarían integradas al nuevo sistema. De esta manera los estudiantes, podrán utilizar estas funcionalidades de una forma más sencilla, ya que harían uso de un sistema gráfico que utiliza metáforas que hacen parte del día a día de las personas, quienes constantemente acceden a aplicaciones en sus dispositivos móviles y computadores. Adicionalmente este aplicativo tendría una nueva funcionalidad que es transparente para el usuario, es decir, que a partir de las credenciales del usuario el sistema adapte las búsquedas de materia y recomiende posibles cursos, que el estudiante podría inscribir para el ciclo académico sin que este suministre datos adicionales.

2. Conclusiones y trabajo futuro

Para concluir, se puede decir que se cumplió con el objetivo general del proyecto de desarrollar una aplicación web para planear versiones tentativas de horario académico debido al alcance del proyecto y la cantidad de horas-hombre trabajadas. Por otro lado, se logró implementar todas las funcionalidades propuestas a pesar de todos los problemas que fueron surgiendo en dicho desarrollo, además, también se cumplieron con todos los objetivos específicos, del mismo modo, teniendo en cuenta las pruebas de usabilidad se puede decir que la solución que nosotros planteamos para resolver la problemática explicada es la más óptima y es mejor que la actual debido a la información recopilada por los usuarios de prueba que llenaron los formularios logrando así sacar estadísticas respecto a lo anteriormente mencionado.

Para terminar, el trabajo futuro por parte de la aplicación JPlanner es mejorar el algoritmo de recomendación de materias, debido a que esta en su primera versión y aunque en las pruebas haya dado un buen resultado puede ser mucho mejor, por otro lado, es muy importante que se implemente la parte de seguridad de la aplicación para que así el single sign-on funcione y se pueda hacer desde MAX.

También es importante mostrar las materias que fueron inscritas y las que no en el calendario, implementar la funcionalidad de dar de baja a una determinada materia, crear un panel de términos y condiciones de la aplicación y por último agregar en los filtros un botón para buscar solo materias recomendadas por el algoritmo anteriormente dicho.

Por otro lado, es necesario decir que el trabajo futuro de la DTI es implementar los servicios web faltantes que no fueron entregados a tiempo para que así desde JPlanner estos puedan ser consumidos.

IX- REFERENCIAS

[1] «La importancia de un diseño web responsive – Internet Ya». [En línea]. Disponible en: <https://www.internetya.co/la-importancia-de-un-diseno-web-responsive/> [Accedido: 19-oct-2018]

[2] «Responsive Web Design: Enriching the User Experience – Website design». [En línea]. Disponible en: <http://www.webdesignblog.gr/wp-content/uploads/2012/03/5.pdf#page=16/> [Accedido: 19-oct-2018]

[3] «The Top Web Development Frameworks in 2018 – JetRuby». [En línea]. Disponible en: <https://expertise.jetruby.com/the-top-web-development-frameworks-in-2018-b31dc7263875> [Accedido: 19-oct-2018]

[4] Learning, C. (s.f.). *Civitas Learning*. Recuperado el 22 de 10 de 2018, de Civitas Learning: <https://www.civitaslearning.com/careers/>

[5] U, Y. (s.f.). *York U*. Recuperado el 22 de 10 de 2018, de York U: <https://registrar.yorku.ca/enrol/guide/vsb>

[6] Oracle. (19 de 5 de 2019). docs.oracle.com. Obtenido de <https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html>

[7] Angular. (19 de 5 de 2019). Angular.io. Obtenido de <https://devdocs.io/angular/>

[8] ISO/IEC/IEEE 42010:2011. Systems and software engineering - Architecture description.

[9] IEEE 1016:2009. Standard for Information Technology - Systems Design - Software Design Descriptions.

[10] RUP. Software Architecture Document.

http://sce.uhcl.edu/helm/RUP_school_example/wcsoftwareprocessweb/templates/ana_de_si/pt_sad.htm

[11] Kruchten, P. Architectural Blueprints—The “4+1” View Model of Software Architecture.
<http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf>

[12] Ambler, S. UML Data Modeling Profile.

<http://www.agiledata.org/essays/umlDataModelingProfile.html>

[13] Ambler, S. User Interface Flow Diagrams.

<http://www.agilemodeling.com/artifacts/uiFlowDiagram.htm>

[14] Ecured. Diagrama Entidad Relación

https://www.ecured.cu/Diagrama_entidad_relaci%C3%B3n

[15] CYTA. Base De Datos Relacional

http://www.cyta.com.ar/elearn/bd/2_presentaciones_1.htm

[16] Wikifoundry. Ingeniería De Software Furps

<http://clases3gingsof.wikifoundry.com/page/FURPS>

[17] Paszniuk. Json Web Token

<https://www.programacion.com.py/varios/que-es-json-web-token-jwt>

[18] Culturación. Diagrama De Clases

<https://culturacion.com/que-es-un-diagrama-de-clases/>

[19] uml-diagrams.org. state-machine

<https://www.uml-diagrams.org/state-machine-diagrams.html>

[20] uml-diagrams.org. state-machine

<https://www.uml-diagrams.org/deployment-diagrams-overview.html>

[21] INDIA, S. (16 de 08 de 2018). *medium.com*. Obtenido de <https://medium.com/front-end-weekly/react-vs-angular-vs-vue-js-a-complete-comparison-guide-d16faa185d61>

[22] Satrom, B. (2018). *telerik.com*. Obtenido de https://www.telerik.com/docs/default-source/whitepapers/telerik-com/choose-the-right-javascript-framework-for-your-next-web-application_whitepaper.pdf