

Universidad Simón Bolívar  
Departamento de Computación y Tecnología de la Información  
Ingeniería del Software I  
Trimestre Abril-Julio 2018

**INFORME SOBRE SISTEMA DE INSCRIPCIÓN DE  
POSTGRADO:  
SPRINT 1**

**Grupo:** Es Magia Negra

Angélica Acosta 14-10005

Aurivan Castro 14-10205

Ian Goldberg 14-10406

Elvin Quero 14-10869

Manuel Rodriguez 13-11223

Giulianne Tavano 13-11389

Sandra Vera 14-11130

Sartenejas, 31 de mayo de 2018

# INTRODUCCIÓN

En este trabajo se explica cómo fue el proceso de desarrollo del sprint 1 del proyecto de “Sistema de Inscripción de Postgrado”, utilizando Eclipse como ambiente de desarrollo y el framework Django para aplicaciones. Además se habla de la experiencia del proceso de desarrollo aplicando programación por pares entre los miembros del equipo, lo cual permitió trabajar de forma rápida y eficazmente.

El proyecto consiste en la realización de una interfaz que le permita a los coordinadores de postgrados gestionar la oferta de asignaturas y a los estudiantes de postgrado gestionar el proceso de inscripción. En el sprint 1, nos centramos en la primera época del proyecto, que se basa en permitir a los coordinadores de Postgrado, gestionar las asignaturas de sus Coordinaciones para ofrecer un servicio de mejor calidad. Se implementa el desarrollo guiado por pruebas de software (TDD), donde para cada prueba se crea el respectivo código, que permite pasarla. A través de este método, se conoce que las funciones implementadas para el manejo de las asignaturas y la validación de sus datos cumplen con los requisitos y se agiliza el proceso de creación de un código limpio.

La finalidad de esta actividad es familiarizarse con la práctica de desarrollo ágil, conocida como Scrum. Al trabajar en cualquier proyecto en equipo, es importante tener un control de las versiones y una buena organización de las ramas, a fin de que los miembros puedan desarrollar de forma amena. Como se implementó la práctica de la programación por pares, el respeto, la buena comunicación y conocer las fortalezas de los miembros, es fundamental para alcanzar los objetivos planteados.

El informe constó de 4 capítulos, se inicia en el capítulo I con la explicación de las historias de usuarios a implementar y su correspondiente análisis de dominio. En el capítulo II, se habla de cómo fue el trabajo en equipo, se muestra con una tabla como fue la distribución de los labores y cómo fue la experiencia el proceso de programación por pares para el equipo. Para el capítulo III, se señala cómo fue la implementación o experiencia con TDD, la cual se viene aplicando desde las asignaciones anteriores. En el cuarto capítulo, se señala como fue el manejo del desarrollo en GitHub. A modo de conclusión, se dan recomendaciones generales para futuros trabajos en equipo.

## CAPÍTULO I

### ANÁLISIS DE HISTORIAS DE USUARIOS

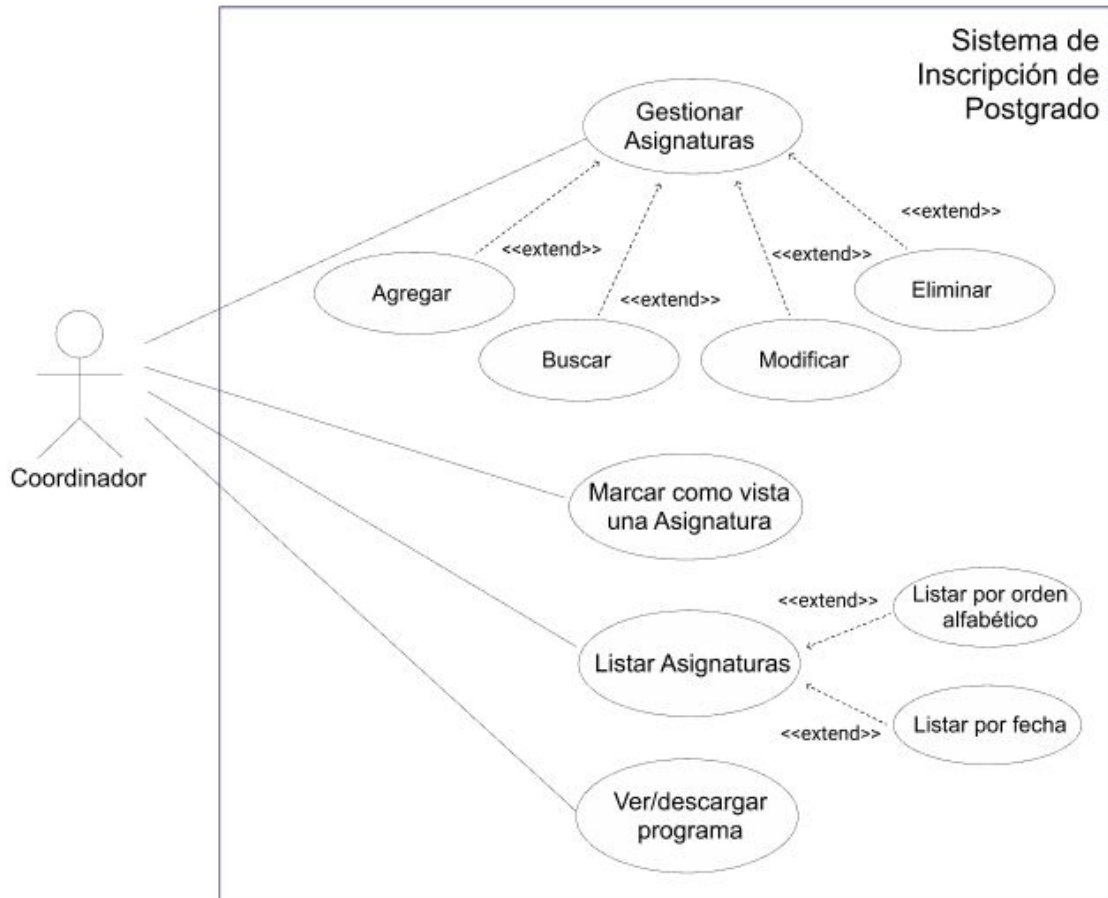


Diagrama de casos de uso del Sprint 1

La épica épica para este primer sprint es la siguiente: ***Como Coordinador de Postgrado, puedo gestionar las asignaturas de mi Coordinación de Postgrado para ofrecer un servicio de mejor calidad.***

Inicialmente, fueron creadas las bases de datos para Coordinación y Asignatura para el cumplimiento de lo requerido en la épica. A continuación se analizará el dominio de datos de los campos para cada tabla.

La tabla correspondiente a la base de datos de Coordinación presenta los campos código de coordinación (“Cod\_coordinacion”) y nombre de la coordinación (“Nombre\_coordinacion”). El dominio del código de la coordinación consiste en dos letras

mayúsculas, lo cual coincide con las fronteras del dominio. Por otra parte, las esquinas del dominio son códigos de coordinación que posean tres letras mayúsculas o minúsculas, o una letra mayúscula o minúscula. El caso malicioso para este dominio fue un código con dos letras en minúscula. En relación al dominio del campo de nombre de la coordinación, consiste en un string de letras que puede tener espacios, por lo cual las fronteras son un string de mínimo un caracter o máximo 30 caracteres, las esquinas son un string vacío o un string que sobrepasa la longitud máxima y los casos maliciosos corresponden a nombres con números.

La tabla de Asignatura posee los campos código de la asignatura (“Cod\_asignatura”), nombre de la asignatura (“Nombre\_asig”), código de la coordinación a la que pertenece la asignatura (“Cod\_coordinacion”), cantidad de créditos que corresponden a dicha asignatura (“Creditos”), la fecha de modificación de la asignatura (“Fecha”), un booleano que indica si la materia fue vista o no (“Visto”) y un campo para el programa de la asignaturas (“Programa”). El dominio para el campo de código de asignatura consiste en un par de letras en mayúscula y cuatro dígitos, es decir que esto representa las fronteras, por lo que las esquinas son menos o más letras mayúsculas y menos o más números con sus combinaciones; y los casos maliciosos son letras minúsculas en vez de mayúsculas.

Para el campo de nombre de la asignatura, de manera análoga al nombre de la coordinación, el dominio son strings de letras que puede contener espacios, las fronteras son un string de mínimo un caracter o máximo 30 caracteres, las esquinas son un string vacío o un string que sobrepase la longitud máxima y los casos maliciosos corresponden a nombres con números. En cuanto al campo de código de coordinación, debido a que es una clave foránea a la tabla de Coordinación, el dominio coincide con el dominio del campo código de coordinación de dicha tabla analizado anteriormente. El dominio del campo de cantidad de créditos son los enteros mayores a cero y menores o iguales a 30, las esquinas son los enteros menores o iguales a cero y mayores que 30. Los casos maliciosos consisten en números con letras y únicamente letras.

Además, se tiene que el dominio del campo Fecha es la fecha correcta de su creación. Para el atributo Visto, al ser booleano solo es válido los valores True o False, si se intenta agregar otro valor, daría error. Por último, al campo Programa solo acepta urls que deben

redirigir al pdf del programa de la materia, el link puede ser modificado. Por otra parte, se tiene que ninguno de los campos de texto acepta acentos, ni ñ.

Una vez dividida la épica en historias de usuario más específicas, se definieron las siguientes tareas:

- ***Como Coordinador de Postgrado, puedo agregar asignaturas a mi Coordinación de Postgrado para gestionar las asignaturas.***

El dominio de los datos corresponde a las tablas de coordinación y asignaturas. Como ambas entidades se relacionan por el campo o atributo “Cod\_coordinacion”, podemos, estando en la vista de la Coordinación, agregar nuevas materias que pertenezcan a la coordinación en cuestión.

- ***Como Coordinador de Postgrado, puedo buscar asignaturas en mi Coordinación de Postgrado para gestionar las asignaturas.***

El dominio para esta historia de usuario está basado en las tablas de coordinación y asignaturas, las cuales se relacionan mediante el atributo o campo correspondiente al código de la coordinación, por lo cual es posible estando en la vista de la Coordinación buscar las asignaturas agregadas por coordinador. Además, es posible filtrar los resultados usando el campo correspondiente al nombre de la asignatura. Por tanto, el dominio para esta historia consiste específicamente en los campos anteriormente mencionados.

- ***Como Coordinador de Postgrado, puedo modificar asignaturas de mi Coordinación de Postgrado para gestionar las asignaturas.***

Como dominio para la historia de usuarios presente se tienen las tablas de coordinación y asignaturas. En este caso, dado que se quieren modificar las asignaturas creadas en la coordinación, gracias al atributo del código de coordinación en la tabla de asignatura, se puede modificar una asignatura desde la vista de la coordinación. En este caso el dominio es el especificado para todos los campos de la tabla asignatura, a excepción del código de la materia, puesto que es la clave de la tabla en la base de datos.

- ***Como Coordinador de Postgrado, puedo eliminar asignaturas de mi Coordinación de Postgrado para gestionar las asignaturas.***

Para esta historia de usuarios, al igual que las anteriores se tiene como dominio las tablas de coordinación y asignaturas. Estando en una coordinación en específico se puede observar las asignaturas que pertenecen a ella y por medio de un botón, se puede eliminar la asignatura seleccionada del registro.

- ***Como Coordinador de Postgrado, puedo listar por orden alfabético (ascendente, descendente) o por fecha de ejecución de las asignaturas de mi Coordinación de Postgrado concreta para gestionar las asignaturas.***

Como dominio para esta historia de usuarios se tienen las tablas de coordinación y asignaturas, desde mi coordinación puedo observar la lista de las materias que pertenecen a ella, estas con una opción se pueden ordenar alfabéticamente (descendente o ascendente) por el nombre. Además como en la tabla de las materias tienen el atributo de “Fecha”, podemos ordenarlos según este atributo.

- ***Como Coordinador de Postgrado, puedo marcar como vista una asignatura de mi Coordinación de Postgrado para gestionar mis asignaturas.***

Al igual que las historias anteriores se tiene que el dominio de esta historia abarca las tablas de coordinación y asignaturas, en específico al campo “Visto” de la tabla asignaturas. Al agregar una nueva materia, se puede hacer “check” indicando que ha sido vista o no, se puede modificar después en edificar.

- ***Como Coordinador de Postgrado, puedo ver y descargar el programa de una asignatura de mi Coordinación de Postgrado para gestionar mis asignaturas.***

El dominio de esta historia de usuarios abarca las tablas de coordinación y asignaturas, específicamente el campo de “Programa”, el cual contiene un url correspondiente al programa de la asignatura. La tabla de Asignatura nos permite ver dicho url y, al ingresar, se presenta la opción de descargarlo.

## CAPÍTULO II

### TRABAJO EN EQUIPO

El trabajo se dividió en parejas, permutándolas a lo largo del desarrollo para permitirle a cada miembro del equipo trabajar con diferentes personas, para así brindarle a las participantes de esta actividad una experiencia variada con respecto a la programación por pares. Asimismo, dicha permutación permitió que el equipo completo se integrará en tareas de desarrollo variadas. A cada par se le asignó una o dos historias de usuario por épica. Se conoce que fueron aproximadamente 10 horas por cada miembro del equipo.

La distribución de los labores se muestra a continuación:

Historia	Tareas	Dueño
Como Coordinador de Postgrado, puedo agregar asignaturas a mi Coordinación de Postgrado para gestionar las asignaturas.	Diseñar la base de datos	Acosta y Tavano
	Diseñar la interfaz del usuario	Quero y Vera
	Implementar el código del back-end	Castro y Rodríguez
	Implementar el código del front-end	
	Completar las pruebas unitarias	Acosta y Quero
	Completar las pruebas de regresión	
	Escribir la documentación	Acosta, Goldberg y Vera
Como Coordinador de Postgrado, puedo buscar asignaturas en mi Coordinación de Postgrado para gestionar las asignaturas.	Implementar el código del back-end	Goldberg y Acosta
	Implementar el código del front-end	
	Completar las pruebas unitarias	Tavano y Quero
	Completar las pruebas de regresión	
Como Coordinador de Postgrado, puedo modificar asignaturas de mi Coordinación de Postgrado para gestionar las asignaturas.	Implementar el código del back-end	Vera y Quero
	Implementar el código del front-end	
	Completar las pruebas unitarias	Acosta y Castro
	Completar las pruebas de regresión	
Como Coordinador de Postgrado, puedo eliminar asignaturas de mi Coordinación	Implementar el código del back-end	Rodríguez y Quero
	Implementar el código del front-end	
	Completar las pruebas unitarias	Tavano y Goldberg

de Postgrado para gestionar las asignaturas.	Completar las pruebas de regresión	
Como Coordinador de Postgrado, puedo listar por orden alfabético (ascendente, descendente) o por fecha de ejecución de las asignaturas de mi Coordinación de Postgrado concreta para gestionar las asignaturas.	Implementar el código del back-end	Rodríguez y Quero
	Implementar el código del front-end	
	Completar las pruebas unitarias	Tavano y Goldberg
	Completar las pruebas de regresión	
Como Coordinador de Postgrado, puedo marcar como vista una asignaturas de mi Coordinación de Postgrado para gestionar mis asignaturas.	Implementar el código del back-end	Goldberg y Castro
	Implementar el código del front-end	
	Completar las pruebas unitarias	Vera y Acosta
	Completar las pruebas de regresión	
Como Coordinador de Postgrado, puedo ver y descargar el programa de una asignatura de mi Coordinación de Postgrado para gestionar mis asignaturas.	Implementar el código del back-end	Vera y Goldberg
	Implementar el código del front-end	
	Completar las pruebas unitarias	Rodríguez y Castro
	Completar las pruebas de regresión	

Tabla 1: Tabla de distribución de labores.

En cuanto a la experiencia de trabajo por pares, consideramos que fue amena y enriquecedora, pues pudimos complementar nuestros conocimientos y trabajar en equipo. Gracias a la buena comunicación y organización, se logró llevar a cabo la implementación de las tareas de una forma más rápida, eficiente y llevadera, manteniendo en todo momento una actitud de apoyo y compañerismo entre los miembros del equipo.



## **CAPÍTULO III**

### **IMPLEMENTACIÓN CON TDD**

La programación dirigida por casos de prueba es una metodología que permite asegurar el correcto funcionamiento del código de forma progresiva. El enfoque TDD consiste en la realización de casos de prueba con características específicas y la posterior modificación del código para que pase correctamente las pruebas de aceptación creadas. Esto se hace con el fin de que se tomen en consideración todos los tipos de entrada posible mientras se desarrolla el proyecto y así disminuir la probabilidad de que ocurra un error durante la ejecución del programa, lo que lleva a un fallo del software.

Para el diseño de los casos de prueba, se utilizó la interfaz que brinda Django, que es una extensión de las pruebas con unittest. Se diseñaron pruebas para comprobar las validaciones del dominio de todos los campos de las tablas de la base de datos, así como también las consultas realizadas en la base de datos.

Previamente se había trabajado con el enfoque TDD, y en este caso las experiencias fueron similares. Realizar el análisis del dominio para luego llevar a cabo las pruebas de las validaciones correspondientes para cada campo de las tablas de Asignatura y Coordinación aseguró la confiabilidad del producto final.

## **CAPÍTULO IV**

### **MANEJO DEL REPOSITORIO GITHUB**

El manejo del repositorio GitHub se realizó de la siguiente manera. Se tuvo en primer lugar una rama primaria «master», al momento de comenzar el desarrollo de software se crearon una serie de ramas. La primer llamada «base\_datos», ahí se desarrolló la base de datos correspondiente al modelo solicitado. Otra rama, llamada «feature/agregar\_asignatura» donde paralelamente se iban implementando las vistas que iban a ser usadas.

Una vez realizada la base de datos, se unieron las ramas de «feature/agregar\_asignatura» y la de «base\_datos». Otra rama de nombre «feature/buscar\_asignatura» se iba implementando el back-end correspondiente a la función de buscar y administrar las asignaturas. Al momento de hacer la integración con las otras ramas, se hizo en una rama «develop».

Para las pruebas unitarias se tiene una rama llamada «feature/test» donde se prueba el buen funcionamiento de la aplicación, en esta rama se implementaron todos los casos sobre las tablas coordinación y asignaturas. Debido a que luego se generaron otros cambios en la base de datos, se tuvo que crear otra llamada «develop\_test». Cada vez que se comprobaba que las ramas tenían correcto funcionamiento, se hacía un merge con «develop» para mantener actualizada la rama del desarrollo.

## CONCLUSIÓN

El objetivo principal de esta actividad fue proveer a los coordinadores de Postgrado una herramienta con la cual gestionar las asignaturas de su coordinación. Para lo antes mencionado se comenzó realizando un análisis de dominio para el problema propuesto basado en la primera historia de usuario descrita en el Product Backlog. Una vez obtenido los resultados del análisis se tomó un esquema de trabajo basado en TDD, apoyándonos en el software de control de versiones Git, con lo cual se pudo coordinar de buena manera las tareas realizadas por los desarrolladores, y en la herramienta de pyunit integrada a eclipse, la cual nos permitió llevar a cabo la revisión de los casos de prueba.

Para cada uno de los casos hallados con el análisis de dominio se creó una función que sirve para verificar o “testear” que ese caso esté contemplado en la solución del problema. La base de datos del proyecto fue desarrollada en su totalidad, sin embargo en este sprint los casos de prueba estuvieron dirigidos a las operaciones de los coordinadores y la gestión de asignaturas, para asegurar el correcto funcionamiento de las historias de usuarios. Se implementó el backend y el frontend que permite al coordinador añadir, modificar, eliminar y consultar asignaturas.

En la práctica Scrum es necesaria una buena planificación de las tareas, lo que se logra con el apoyo de un líder que guíe al equipo y asigne las actividades dependiendo de las habilidades que posea cada miembro. La comunicación efectiva y un ambiente de trabajo armónico es fundamental para poder desarrollar satisfactoriamente las funcionalidades y requisitos del proyecto. Por otro lado, en lo que concierne a la realización de casos de prueba, es importante tomarse el tiempo para analizar el dominio de los casos de prueba, pues así podemos garantizar la completitud y confiabilidad del código.