

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 3**

**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, 27 de junio de 2018

## INTRODUCCIÓN

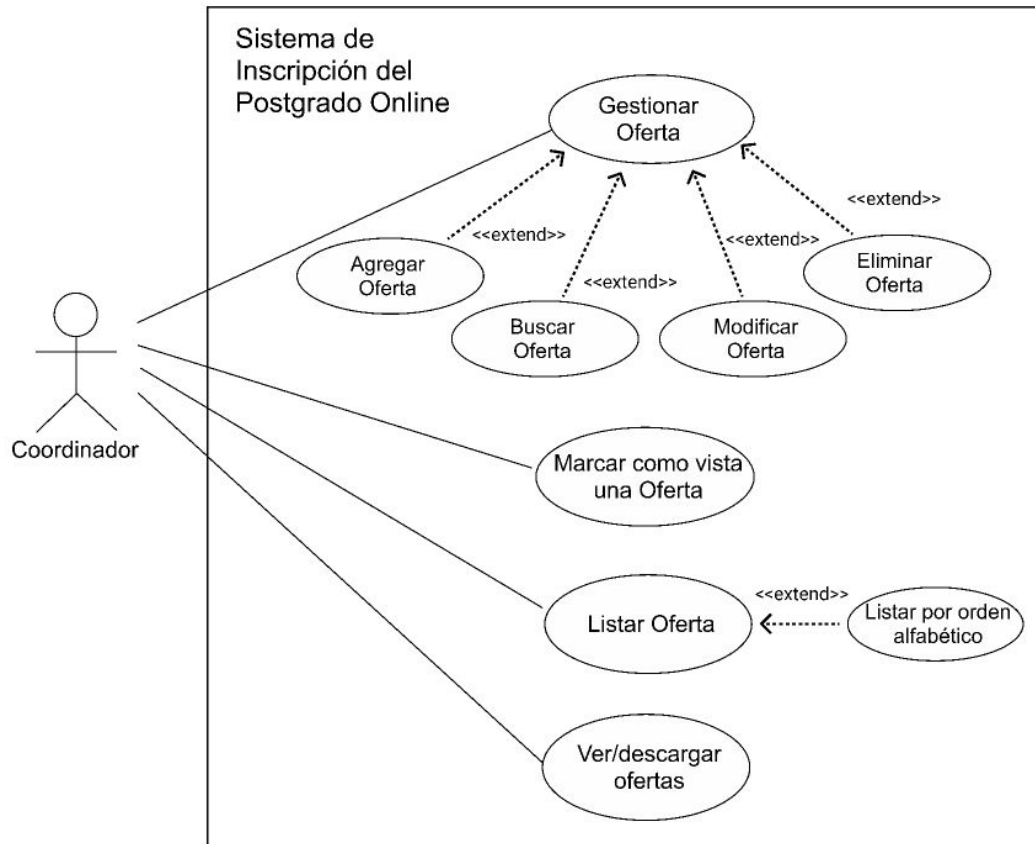
El presente trabajo describe el proceso de desarrollo del sprint 3 del proyecto de “Sistema de Inscripción de Postgrado”, utilizando el framework Django para aplicaciones, cubriendo todos los aspectos desde la planificación a través de las historias de usuario hasta la implementación del software y el manejo del repositorio empleado.

El sprint 3 se centró en la segunda épica del proyecto, la cual abarca la implementación de las funcionalidades con las cuales cada Coordinador de Postgrado, puede “postular asignaturas a una oferta de asignaturas específica para gestionar mis ofertas académicas trimestrales”. Se inició obteniendo el análisis de dominio y posteriormente, mediante su uso, se crearon los casos de prueba requeridos para la tabla de Trimestre y se modificaron los de la tabla de Se Ofrece implementada en el sprint anterior. Mediante tests se busca verificar que cada uno de los requisitos para las funcionalidades que involucran esas tablas son tomados en cuenta y que las validaciones presentes en sus campos son hechas de forma correcta.

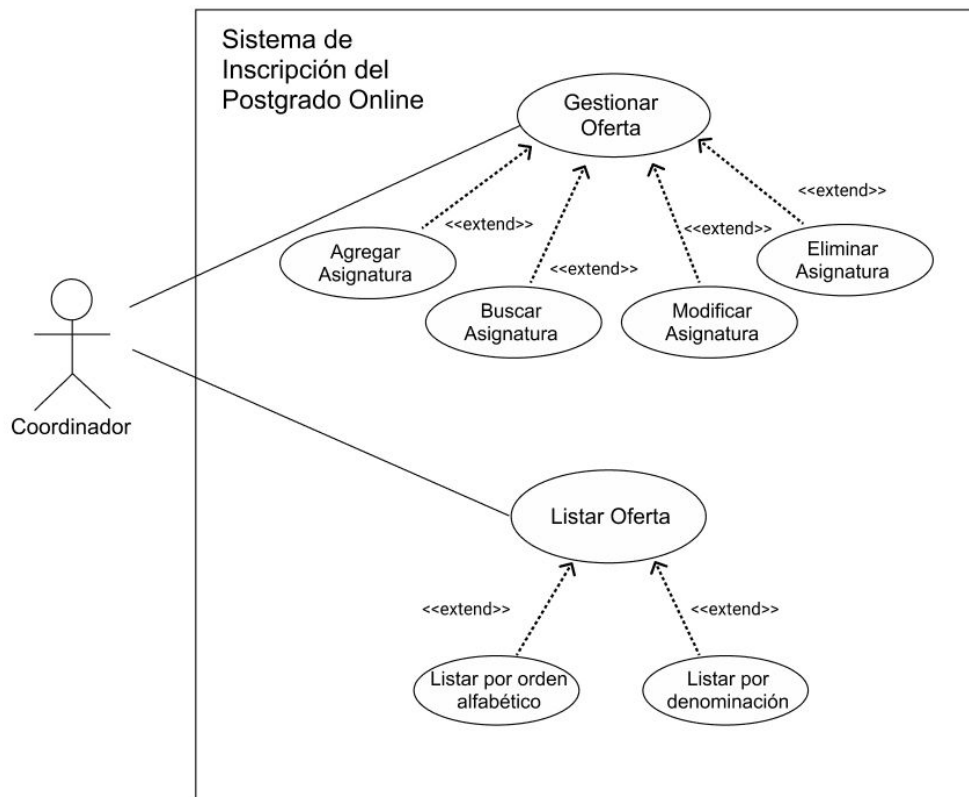
El presente informe se dividió en tres capítulos. En el primero se describen las distintas historias de usuario que componen la épica desarrollada para este sprint y su respectivo análisis de dominio. En el segundo capítulo se habla sobre la organización, a través de una tabla de distribución de carga, y la experiencia de trabajar en equipo. Por último en el tercero se muestra el manejo del repositorio de GitHub llevado a cabo por el equipo. Finalmente se concluye con un breve resumen sobre el proceso de desarrollo del software y se colocan algunas recomendaciones para futuros trabajos en equipo, y además se incluye en los anexos una tabla con el flujo básico del software desarrollado.

# CAPÍTULO I

## ANÁLISIS DE DOMINIO



Caso de uso - Sprint 2.



#### Caso de uso - Sprint 3.

La épica para el sprint es la siguiente: ***Como Coordinador de Postgrado, puedo postular asignaturas a una oferta de asignaturas específica para gestionar mis ofertas académicas trimestrales.***

Para llevar a cabo las funcionalidades que comprende la épica fue agregada la tabla de “Trimestre” y se modificó la tabla de “Se\_Ofrece”.

#### Trimestre:

Esta tabla posee una clave compuesta por una tupla, la cual está formada por un campo llamado “Periodo” y otro llamado “Año”. Esto corresponde a los periodos de ofertas.

Para el caso de “Periodo” tenemos que admite strings con una longitud máxima de siete caracteres y que la información introducida en este campo se encuentre en la lista (‘ENE-MAR’, ‘ABR-JUL’, ‘VERANO’, ‘SEP-DIC’), en la interfaz el usuario sólo podrá escoger un elemento dentro de dicha lista.

Para el caso de “Anio” tenemos que admite un número entero que se comprende entre 1970 y un año posterior al año actual. Debido a las características anteriores el análisis de dominio es el siguiente:

Los casos de esquina, o frontera ya que hablamos de una sola variable, corresponden a los bordes del intervalo admitido, los cuales son 1970 y, para este año, 2019, por lo que los casos de prueba de esquina serían éstos. Para los casos de malicia sería colocar un año anterior a 1970, es decir 1969, y uno posterior al año actual más uno, es decir, 2020

### **Se\_ofrece:**

Corresponde a las asignaturas ofertadas en un periodo de oferta específico. Tiene una referencia a la tabla descrita arriba.

Tiene un campo en el que se introduce el id del profesor que impartirá la asignatura que se ofrece, el cual hace referencia al campo “id\_prof” que se encuentra en la tabla “Profesor”, por lo que comparten el análisis de dominio.

El campo “Cod\_asignatura” hace referencia a un campo con el mismo nombre ubicado en la tabla “Asignatura”, y comprende el mismo análisis de dominio que aquella.

La tabla consta con un campo llamado “Horario” el cual debe constar de dos números enteros separados por un guión (-). La restricción del campo verifica que en efecto lo anterior se cumpla, que el primer número sea menor que el siguiente, que el menor sea positivo y que el mayor no se exceda de 14. La primera hora la tomamos como la primera variable en el análisis y la siguiente como la segunda.

Debido a lo antes mencionado el análisis de dominio está compuesto por los siguientes casos: Las esquinas son los valores marcados en azul, y las fronteras son valores entre las esquinas, de la forma 1-yy, xx-13, xx-yy donde se cumple que  $yy = xx + 1$ . En cuanto a la malicia serían valores donde xx o yy sea 0 o 14.

Para el campo “Dia” debe ser un string, este debe encontrarse en una lista de opciones entre las cuales se encuentran cada uno de los días de la semana a excepción del domingo. En

la interfaz aparece como una lista desplegable por lo que el usuario no podrá elegir valores inválidos.

El campo de “Período” hace referencia a un campo con el mismo nombre en la tabla de “Trimestre”, cuyo análisis de dominio se hizo previamente en este informe.

Por último se cuenta con un campo llamado “Cod\_coordinacion” el cual hace referencia a un campo llamado de igual manera en la tabla de “Coordinacion”. Ambos campos poseen el mismo análisis de dominio.

## **CAPÍTULO II**

### **TRABAJO EN EQUIPO**

Para la realización de estas historias de usuario, continuamos que la organización del trabajo por pares, donde la relación existente es de “experto - novato”. De esta manera, se crea mayor equilibrio entre las distribución de las tareas y permite que los novatos, reforzar los conocimientos obtenidos que en el proceso del desarrollo de la aplicación. En el caso de los expertos, les brinda la oportunidad de compartir sus conocimientos para que todos como equipo puedan incluirse y contribuir en el proyecto.

Para permitir que cada miembros del equipo trabaje con diferentes personas, se intercambiaron las parejas, esto permite que todos tengan una experiencia variada con respecto a la programación por pares. Se conoce que fueron aproximadamente 5 horas por cada miembro del equipo, a cada par se les asignaron diversas tareas por historias de usuario. De esta manera, conocer las fortalezas de los miembros, es fundamental para alcanzar los objetivos planteados.

La distribución de los labores se muestra a continuación:

Historia	Tareas	Dueño
Como Coordinador de Postgrado, puedo agregar una oferta de asignaturas para gestionar mis ofertas académicas trimestrales.	Diseñar la interfaz de usuario	Acosta y Tavano
	Implementar el código del back-end	Castro y Rodriguez
	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 ofertas de asignaturas para gestionar mis ofertas académicas trimestrales.	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	
Como Coordinador de Postgrado, puedo modificar ofertas de asignaturas para gestionar mis ofertas académicas trimestrales.	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 ofertas de asignaturas para gestionar mis ofertas académicas trimestrales.	Implementar el código del back-end	Rodriguez 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 listar por orden alfabético (ascendente, descendente) o por fecha de ejecución de las ofertas de mi Coordinación de Postgrado para gestionar las ofertas académicas trimestrales.	Implementar el código del back-end	Rodriguez 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 ver y descargar las ofertas de asignaturas en un periodo de tiempo estipulado de mi Coordinación de Postgrado para gestionar las ofertas académicas trimestrales.	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	

Tabla 1: Tabla de distribución de labores.

La experiencia con el trabajo por pares fue enriquecedora, debido a que se mantuvo en todo momento una actitud positiva, de apoyo y compañerismo entre los miembros del equipo.

Se tiene que se logró reforzar los conocimientos obtenidos a lo largo del proyecto y por medio de la distribución, se alcanzó agilizar el proceso de desarrollo de las tareas, siendo más grato.



### **CAPÍTULO III**

#### **MANEJO DEL REPOSITORIO GITHUB**

Para el manejo del repositorio GitHub se realizó de la siguiente manera, como en la rama «master» y «develop» tenemos los avances del sprint 1 y el sprint 3, se crearon dos ramas a partir de esta: «feature/arreglar\_sprint2» y «feature/arreglar\_sprint3».

En la rama «feature/arreglar\_sprint2», se manejo la parte de las funcionalidades de las vistas, en esta tiene parte del back-end y front-end de la aplicación correspondiente al sprint 2. Se implementan los métodos de agregar, buscar, eliminar y ordenar las ofertas correspondientes a las coordinaciones. También se trabajó en la rama «develop\_test», los casos de prueba o tests de las tablas de los profesores, trimestres y de las ofertas.

Por otra parte, en «feature/arreglar\_sprint3» se realizaron las modificaciones pertinentes a las funciones que manejan las vistas de la aplicación (back-end y front-end). En esta rama, se cambiaron detalles de la entrega pasada, que permitieron la integración de los cambios hechos en la rama «feature/arreglar\_sprint2». Al verificar que las ramas tenían correcto funcionamiento, se hacía un merge con «develop» para mantener actualizada la rama del desarrollo. Al finalizar, se hizo merge con la rama «master» para actualizarla y tener el proyecto integrado en la rama principal del repositorio.

## CONCLUSIÓN

El propósito del sprint fue modificar las funcionalidades del sprint anterior para integrar al sistema de Inscripción de Postgrado las funcionalidades que permiten a los coordinadores, gestionar las ofertas de su coordinación, las cuales se organizan según periodo. El análisis del dominio que se realizó al comenzar el sprint es similar al del sprint 3, con el se logró asegurar que los datos ingresados por el usuario sean correctos.

Para cada uno de los casos presentados en el análisis de dominio realizado se implementó una función que testee tanto los casos esquina, como los frontera y los maliciosos. La base de datos resultante de este sprint es bastante similar a la del sprint anterior, con la diferencia de que la tabla de “Se\_ofrece” se separó en dos tablas distintas, una que conserva el mismo nombre y todos sus atributos, a excepción de “Anio” y “Periodo” y otra tabla llamada “Trimestre” que consta de los campos que no se encuentran presentes en la tabla anterior. Esto se hizo con el fin de independizar la información de la oferta y el trimestre en el que se impartió o se impartirá.

En lo que concierne a la realización de casos de prueba, es importante tomarse el tiempo para analizar el dominio de estos y de esta forma poder garantizar que las funcionalidades generadas por el código produzcan el resultado esperado y de ésta forma el software sea confiable y completo.

Al implementar la práctica de Scrum, se conoce que es fundamental tener un ambiente de trabajo armonioso, donde es fundamental que los miembros del equipo mantengan una buena comunicación y se respeten mutuamente para lograr una buena cohesión entre los miembros que lo componen y de esta forma cumplir con todos los requerimientos y funcionalidades solicitadas.

## ANEXOS

### FLUJO BÁSICO

\* : Código de coordinación

\*\* : Código de asignatura

\*\*\*: ID correspondiente a un elemento de la tabla

\*\*\*\*: Criterio de búsqueda

\*\*\*\*\*: ID correspondiente a un elemento de la tabla “Se\_ofrece”

PATH	MÉTODO	FUNCIONES USADAS (en orden)
/	GET	index __renderViewGET__
/index	GET	index __renderViewGET__
/coordinacion/	GET	__redirectCenter__ index __renderViewGET__
/coordinacion/	POST	__redirectCenter__ index __renderViewPOST__ __modifyDB__
/coordinacion_*/	GET	__redirectCenter__ coordinacion __renderViewGET__
/coordinacion_*/	GET	__redirectCenter__ coordinacion __renderViewGET__ __modifyDB__
/coordinacion_CO/edit_**	GET	__redirectCenter__ coordinacion editAsignatura
/coordinacion_*/edit_**	POST	__redirectCenter__ coordinacion

		editAsignatura updateAsignatura
/coordinacion_*/delete_**	GET	__redirectCenter__ coordinacion deleteAsignatura
/coordinacion_*/orderby_No mbre_asig=asc	GET	__redirectCenter__ coordinacion orderbyAsignatura
/coordinacion_*/orderby_No mbre_asig=desc	GET	__redirectCenter__ coordinacion orderbyAsignatura
/coordinacion_*/orderby_Fe cha=asc	GET	__redirectCenter__ coordinacion orderbyAsignatura
/coordinacion_*/orderby_Fe cha=desc	GET	__redirectCenter__ coordinacion orderbyAsignatura
/coordinacion_*/searchInd	GET	__redirectCenter__ coordinacion searchAsignatura
/coordinacion_*/searchInd	POST	__redirectCenter__ coordinacion searchAsignatura
/coordinacion_*/search_No mbre_asig=****	GET	__redirectCenter__ coordinacion searchAsignatura
/coordinacion_*/search_Cod _asig=****	GET	__redirectCenter__ coordinacion searchAsignatura
/periodos/	GET	__redirectCenter__ periodo
/periodos/	POST	__redirectCenter__ periodo __modifyDB__
/periodos/edit_***	GET	__redirectCenter__ periodo editPeriodo

/periodos/edit_***	POST	__redirectCenter__ periodo editPeriodo updatePeriodo
/periodos/orderby_Periodo=asc	GET	__redirectCenter__ periodo orderByPeriodo
/periodos/orderby_Periodo=desc	GET	__redirectCenter__ periodo orderByPeriodo
/periodos/orderby_Trimestre=asc	GET	__redirectCenter__ periodo orderByPeriodo
/periodos/orderby_Trimestre=asc	GET	__redirectCenter__ periodo orderByPeriodo
/periodos/searchInd	GET	__redirectCenter__ periodo searchPeriodo
/periodos/searchInd	POST	__redirectCenter__ periodo searchPeriodo
/periodos/searchInd	POST	__redirectCenter__ periodo searchPeriodo
/periodos/search=****	GET	__redirectCenter__ periodo searchPeriodo
/periodos/search=****/printPdf	GET	__redirectCenter__ periodo searchPeriodo printPdf
/periodos/printPdf	GET	__redirectCenter__ periodo printPdf
/ofertas_***	GET	__redirectCenter__ ofertas
/ofertas_***	POST	__redirectCenter__

		ofertas __modifyDB__
/ofertas_***/edit_*****	GET	__redirectCenter__ ofertas editOferta
/ofertas_***/edit_*****	POST	__redirectCenter__ ofertas editOferta updateOferta
/ofertas_***/delete_*****	GET	__redirectCenter__ ofertas deleteOferta
/ofertas_***/orderby_Cod_a signatura=asc	GET	__redirectCenter__ ofertas orderbyOferta
/ofertas_***/orderby_Cod_a signatura=desc	GET	__redirectCenter__ ofertas orderbyOferta
/ofertas_***/orderby_Nomb re_asig=asc	GET	__redirectCenter__ ofertas orderbyOferta
/ofertas_***/orderby_Nomb re_asig=desc	GET	__redirectCenter__ ofertas orderbyOferta
/ofertas_***/searchInd	GET	__redirectCenter__ ofertas searchOferta
/ofertas_***/searchInd	POST	__redirectCenter__ ofertas searchOferta
/ofertas_***/search_Nombre _asig=*****	GET	__redirectCenter__ ofertas searchOferta
/ofertas_***/search_Nombre _asig=*****/printPdf	GET	__redirectCenter__ ofertas searchOferta printPdf
/ofertas_***/printPdf	GET	__redirectCenter__

		ofertas printPdf
/profesor/	GET	__redirectCenter__ ndex __renderViewGET__
/superuser/	GET	__redirectCenter__ ndex __renderViewGET__

Tabla No.2 : Flujo básico del software.