



UNIVERSIDAD SIMÓN BOLÍVAR
Decanato de Estudios Profesionales
Departamento de Computación y Tecnología de la Información
CI-3715 Ingeniería del Software I
Septiembre-Diciembre 2017

Informe Técnico: Avanzómetro Sprint 3

Profesor: Alfonso Reinoza

Integrantes:

Miguel Canedo 13-10214
Rafael Cisneros 13-11156
Carlos Pérez 13-11089
Ritces Parra 12-11088
Yezabel Rincón 10-11005
Erick Flejan 12-11555
Andrés Buelvas 13-10184
José Bracuto 13-10173

Sartenejas, 01 de diciembre de 2017

Índice general

1. Introducción	1
2. Historias de Usuario y análisis de dominio	2
2.1. Animación varias cohortes	2
2.2. Granularidad de las gráficas	2
2.3. Gráfico de curvas	3
3. Documentación del software	4
3.1. Diseño	4
3.2. Interfaz	5
3.3. Clases	11
3.4. Métodos	12
3.5. Modelos	14
3.6. Vistas	14
4. Trabajo en equipo	17
5. Experiencia TDD	17
6. Manejo del repositorio de Github	18
7. Conclusiones y recomendaciones	19

Capítulo 1

Introducción.

En este documento se presentarán las historias de usuario del sprint 3 del Avanzómetro. El objetivo de las historias de usuario es la realización de una interfaz que permite mostrar gráficos dinámicos e interactivos sobre los créditos aprobados de los estudiantes de varias cohortes y cambiar la granularidad de dichas graficas.

En el capítulo 2 de este informe hablaremos sobre las historias de usuario, que representan las funcionalidades de la aplicación y otras actividades necesarias para su desarrollo y para cada historia de usuario se explican el análisis de su dominio. En el capítulo 3 se especifica la documentación del software, su diseño, interfaz, sus clases, métodos, modelos y vistas. En el capítulo 4 se explicará la división del trabajo y el esfuerzo realizado por el equipo en la realización de cada actividad. El capítulo 5 presenta el desarrollo de las pruebas TDD realizadas para la aplicación y la experiencia del equipo en este aspecto, finalmente el capítulo 6 explica el manejo del repositorio Github de nuestra organización y como nos ayudó en el desarrollo del software. Finalmente en el capítulo 7 se presenta las conclusiones y recomendaciones para el proyecto.

Capítulo 2

Historias de usuario y análisis de dominio.

El prototipo del Avanzómetro cuenta con 13 historias de usuario para ser desarrolladas en 3 sprints. El tercer sprint consta de 3 historias referentes principalmente a la comparación de varias cohortes en la misma animación, y el cambio de la granularidad de las gráficas.

A continuación se explican las 3 historias de usuario del sprint 3 y sus respectivos análisis de dominio.

2.1. Animación varias cohortes

Se requiere visualizar la animación del flujo de estudiantes de dos o más cohortes. Para el desarrollo de esta historia de usuarios se tienen los siguientes criterios de aceptación:

- a. Las cohortes deben tener colores bien diferenciados.
- b. Las cohortes deben compartir el espacio ocupado por las coordenadas de los intervalos.

El dominio de esta historia de usuario es la cantidad de cohortes que se pueden mostrar en una misma gráfica. Por razones estéticas hemos colocado de 1 a 5 cohortes en una misma gráfica.

2.2. Granularidad de las graficas

Se requiere cambiar la granularidad de los diagramas de barra. Para esta historia de usuario se tienen los siguientes criterios de aceptación:

- a. Se debe poder seleccionar la granularidad y las barras deben estar en concordancia con la base de datos.

El dominio de esta historia de usuario es el valor a seleccionar para la granularidad de las barras. Por razones estéticas, se acordó colocar la granularidad con un mínimo de 5 créditos. El máximo valor para la granularidad es de 240, ya que este es el número de créditos total en la mayoría de las carreras de la universidad.

2.3. Gráfico de curvas

Se requiere la opción de ver el flujo de estudiantes de una o más cohortes en un gráfico de curvas.

- a. Las cohortes deben tener colores bien diferenciados.
- b. Las cohortes deben compartir el espacio ocupado por las coordenadas de los intervalos.

El dominio de esta historia de usuario es la granularidad de los créditos para las barras, que va de 5 a 240; el % en la vertical de la gráfica, que va de 0 a 100%; la cantidad de milisegundos que dura una instantánea para un trimestre específico, que va de 500 a 3000 milisegundos; y la cantidad de cohortes graficadas, que son de 1 a 5 cohortes.

Capítulo 3

Documentación del software

3.1. Diseño

Para la realización del Avanzómetro se utilizaron las siguientes herramientas:

- **Django** como framework para desarrollo web.
- **Python** como lenguaje de programación.
- **PostgreSQL** como manejador de base de datos con *Psycopg2* para compatibilidad con *Python*.
- **Html** como lenguaje de desarrollo web.
- **Javascript** para realizar animaciones y validaciones de las páginas web.
- **D3** y **Dimple** para realizar las gráficas mostradas en el software.
- **Selenium** y **geckoDriver** para la realización de las pruebas en el navegador *Mozilla Firefox*.

Para poder hacer uso del Avanzómetro es necesario tener todas estas herramientas instaladas en el ordenador, a excepción de Selenium y geckodriver.

Para instalar estas librerías se usan los siguientes comandos (se debe tener instalado pip de Python):

```
python -m pip install django --upgrade
python -m pip install psycopg2 --upgrade
python -m pip install selenium --upgrade
```

La instalación de geckodriver se encuentra en el siguiente link:

https://www.obeythetestinggoat.com/book/pre-requisite-installations.html#firefox_gecko

Como estructura de datos se usó Json para tomar los arreglos del controlador de Django y pasarlo al html.

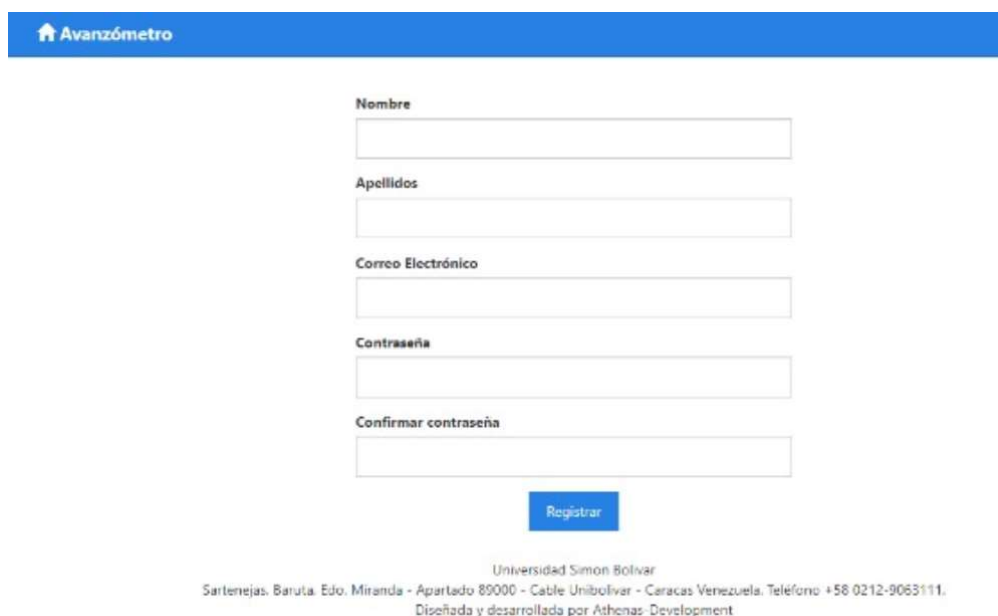
3.2. Interfaces

Al iniciar el avanzómetro se muestra la interfaz de login, aquí se facilitan los campos (inputs) para el nombre de usuario (correo) y la contraseña.



The login interface features a blue header with a home icon and the text 'Avanzómetro'. Below the header, there are two input fields: 'Correo Electrónico' and 'Contraseña'. A link labeled 'Regístrate' with a pencil icon is positioned below the password field. A blue button labeled 'Ingresar' is located at the bottom of the form. At the very bottom, there is a footer containing the text: 'Universidad Simón Bolívar', 'Sartenejas, Baruta, Edo. Miranda - Apartado 89000 - Cable Unibolivar - Caracas Venezuela. Teléfono +58 0212-9063111.', and 'Diseñada y desarrollada por Athenas-Development'.

Además se tiene la opción de registro, que al seleccionarla te redirecciona a la página del registro que cuenta con un formulario de datos para introducir nombres, apellidos, correo y contraseña con los cuales se ingresaran posteriormente al sistema.



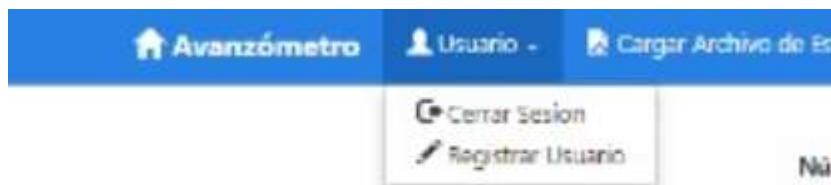
The registration interface features a blue header with a home icon and the text 'Avanzómetro'. Below the header, there are five input fields: 'Nombre', 'Apellidos', 'Correo Electrónico', 'Contraseña', and 'Confirmar contraseña'. A blue button labeled 'Registrar' is located at the bottom of the form. At the very bottom, there is a footer containing the text: 'Universidad Simón Bolívar', 'Sartenejas, Baruta, Edo. Miranda - Apartado 89000 - Cable Unibolivar - Caracas Venezuela. Teléfono +58 0212-9063111.', and 'Diseñada y desarrollada por Athenas-Development'.

Una vez conectado, se muestra la interfaz para generar animaciones, en esta página se observa una primera animación de ejemplo completamente vacía, observándose como los trimestres a la derecha de la gráfica se van seleccionando uno a uno.

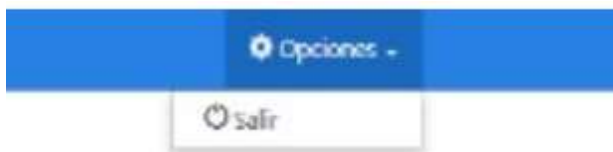


En la parte superior de la página se observa la barra de navegación con 4 botones:

El botón “Usuarios” despliega 2 opciones: una para salir del sistema y otra para registrar otros usuarios.



El botón “Opciones” despliega un menú para ser utilizado en el futuro. Actualmente cuenta con una sola opción usada para salir del sistema.



El botón “Cargar archivo de estudiantes” te redirecciona a la interfaz para cargar archivos. Y el botón “Crear Animación” te envía a la página principal, la misma que aparece al conectarse al sistema.



La interfaz para cargar archivo de estudiantes cuenta con un botón gris “Seleccionar Archivo”, al hacer click sobre él se abre el explorador de archivos para seleccionar el archivo a cargar. Luego se tiene en azul el botón “Cargar”, para subir los datos al sistema.

Usuario - Cargar Archivo de Estudiantes Crear Animación

Seleccione un archivo .csv para cargar Estudiantes:

Documento:

Seleccionar archivo Ningún archivo seleccionado

Cargar

Universidad Simón Bolívar
Sartenejas, Baruta, Edo. Miranda - Apartado 89000 - Cable Unibolivar - Caracas Venezuela. Teléfono +58 0212-9063111.
Diseñada y desarrollada por Athenas-Development

El formato del archivo debe ser .csv y los datos deben estar organizados como en el siguiente ejemplo:

A	B	C	D	E	F	G	H	I
Carnet	Cohorte	Nombre	1ro_1	1ro_2	1ro_3	2do_1	2do_2	2do_3
90-11111	1990	Donato Adolfo Corro	0	0	16	16	32	32
90-22222	1990	Donat Adolfo Corro	16	0	32	16	48	32
90-33333	1990	Dona Adolfo Corro	16	0	32	16	48	32
90-44444	1990	Don Adolfo Corro P	32	16	48	32	64	48
90-55555	1990	Do Adolfo Corro Pe	32	16	48	32	64	48
90-66666	1990	D Adolfo Corro Per	32	32	48	48	64	64

Completando la cantidad de trimestres requeridos según el ejemplo (La columna del primer trimestre del cuarto año de carrera ocuparía la columna 4to_1)

Para crear animaciones, en la interfaz de animaciones (primera página al iniciar sesión, o haciendo click sobre “Crear Animación”) se introduce el número de cohortes a graficar, debe ser un numero entre 1 y 5, y se presiona el botón “Cargar”. Esto nos envía a una nueva interfaz con un formulario para completar datos necesarios para la animación además de la animación ejemplo.

Avanzómetro

Usuario

Cargar Archivo de Estudiantes

Crear Animación

Opciones

Carrera:

Seleccione una

Granularidad:

Seleccione una

Millisegundos por imagen:

Seleccione una

Tipo de Gráfica:

Barra

Linea

Cohorte 1:

Seleccione una

Cohorte 2:

Seleccione una

Cargar Datos

Animación Ejemplo

Carrera: XXXXXXXX

Se selecciona el tipo de graficas a obtener: de barras o líneas; las cohortes a graficar; la carrera deseada; la granularidad de los créditos (entre 5 y 240); y los milisegundos que cada instantánea aparecerá en pantalla (entre 500 y 3000 milisegundos). Al completar los datos se debe hacer click en el botón “Cargar Datos”, y aparece la animación en la parte inferior, con los botones para cambiar las velocidades y pausar/comenzar la animación.



Con el botón “Play” se activa la animación, y con “Pausar” se detiene. Los botones “Aumentar” y “Disminuir” cambian la velocidad de 500 en 500 milisegundos. “Anterior” y “Siguiente” cambian la gráfica actual por la anterior o posterior respectivamente. “Resetear” pausa la animación y la coloca en el primer trimestre. Los botones a la derecha representan cada trimestre, al hacer click sobre ellos se pausa la animación y se muestra la imagen correspondiente al trimestre seleccionado.

3.3. Clases

Las clases creadas por el equipo de desarrollo se explican en la sección 3.6 de este informe. Para el desarrollo del avanzómetro se usó las siguientes clases de Django:

Nombre	Propósito
django.contrib.auth.models	Se usa para generar las tablas de autenticación de usuarios del sistema.
django.views.generic	Esta clase es usada por Django para generar vistas a partir de una clase con el método "CreateView".
django.contrib.auth.views	Contiene las vistas para autenticación de usuarios.
django.conf	Contiene la configuración de Django.
django.contrib.auth.decorators	Esta clase se usa para el control de usuarios en las diferentes páginas del software.
django.shortcuts	Clase que ayuda el manejo y redirección de las páginas web.
django.core.cache	Clase usada para limpiar la memoria cache del sistema cada vez que se corre un método.
json	Objeto que permite transferir datos de la base de datos desde un controlador hacia la vista correspondiente.
SVG	Formato de las gráficas usadas en html.
models.Model	Objeto que permite definir una tabla en la base de datos.

3.4. Métodos

Los métodos creados por el equipo de desarrollo se explican en la sección 3.6 de este informe. En el desarrollo del Avanzómetro se usaron las siguientes funciones:

Nombre	Propósito
RegistroForm	Método creado por el equipo de desarrollo para generar el formulario de registro
render	Método que permite a una función del controlador mostrar una vista correspondiente.
login_required	Decorador de Django para forzar la existencia de un usuario conectado al hacer uso de la función acompañada del decorador.
logout_then_login	Método usado para redireccionar a la página de login una vez se desconecta del sistema.
- json.dumps	Método usado para convertir una matriz de diccionario en un objeto json.
- [Tabla].save	Guarda los datos en la base de datos en la tabla Tabla.
request.POST.get	método usado para obtener variables enviadas de un formulario tipo post
[Tabla].save	Guarda los datos en la base de datos en la tabla Tabla.
[Tabla].filter	Realiza una consulta de la base de datos.
- [Tabla].count	cuenta cuantas entradas hay en la tabla Tabla
getElementById	Método de javascript para obtener una etiqueta html determinada según su id.
dimple.newSvg	Función de la librería “Dimple” que comienza un nuevo lienzo para una animación.

Nombre	Propósito
JSON.parse	Recibe un objeto Json y lo transforma en datos usables en el html.
dimple.filterData	filtra un objeto Json según una llave del diccionario
dimple.chart	Crea una gráfica usando un lienzo svg y un diccionario Json. addEventHandler
indicator.setBounds	Configura el margen de la imagen.
addEventHandler	Función de javascript para crear un manejador de eventos.
addCategoryAxis	Función de dimple que define cual va a ser el eje y de una gráfica.
addMeasureAxis	Función de dimple que define cual va a ser el eje x de una gráfica.
addSeries	Función de dimple configura el tipo de gráfico mostrado.
draw	Función de dimple que redibuja la gráfica con la configuración establecida.
selectAll	Selecciona todos los elementos de la gráfica que coincidan con el parámetro especificado.
setStoryboard	selecciona todos los elementos de la gráfica que coincidan con el parámetro especificado
selectAll	selecciona todos los elementos de la gráfica que coincidan con el parámetro especificado
setStoryboard	Función de dimple que permite hacer una animación con los datos de la gráfica.
addOrderRule	Establece una regla de orden para las gráficas.

3.5. Modelos

La base de datos usada cuenta con 4 tablas para los datos usados en las gráficas: estudiantes, trimestre, cohorte y cursa que es una tabla para relacionar un estudiante con un trimestre y los créditos aprobados por ese estudiante en ese trimestre. Estas tablas se definen en el archivo models.py de la aplicación registro. Para el login y registro de usuarios se usan tablas que provee Django donde se implementan todas las medidas de seguridad referentes a los usuarios.

3.6. Vistas

En cada aplicación se encuentra un archivo llamado views.py que funciona como controlador de Django para el software. A continuación explicaremos el contenido de este archivo en cada aplicación.

En la aplicación “registro” usa la clase RegistroUsuario, esta clase es utilizada por Django para poder crear la vista correspondiente al registro de usuarios.

En la aplicación “login” se encuentran las funciones “index” y “logout”:

Nombre	Propósito
index	Este método direcciona a la página principal del avanzómetro.
logout	Es el método utilizado para desconectar a un usuario del sistema, y lo redirecciona a la página de login del Avanzómetro.

En la aplicación “instantánea” se usan los métodos “getcreditsbytrandct” e “Instantánea”:

Nombre	Propósito
getcreditsbytrandct	Este método consigue en la base de datos los créditos de cada estudiante de una cohorte, según el trimestre señalado, para la generación de instantáneas.
Instantánea	Genera una instantánea básica cuando no tiene especificados datos de entrada, y una instantánea correspondiente a una cohorte según los datos dados.

A continuación se explican los métodos usados por la aplicación “carga” que son “comprobar_entero”, “separar_estudiantes”, “separar_estudiantes_V2”, “read_csv”, u “CargarArchivo”:

Nombre	Propósito
comprobar_entero	Verifica que no se introduzcan notas negativas ni elementos distintos a un entero positivo.
separar_estudiantes	Lee el archivo csv e introduce en la base de datos los estudiantes indicados en el archivo con sus respectivos datos usando el método read_csv. El archivo csv usado por este método es una simulación del expediente de DACE.
separar_estudiantes_V2	Similar al método anterior, lee un archivo csv e introduce en la base de datos los estudiantes con sus respectivos datos usando el método read_csv. En este método el archivo csv contiene datos más sencillos para realizar pruebas sobre las gráficas y animaciones.

Nombre	Propósito
read_csv	Esta función es la encargada de introducir los datos de los archivos csv en la base de datos.
CargarArchivo	Este es el método usado para Django para invocar la vista creada.

A continuación se explican los métodos usados por la aplicación “animación” que son “obtenerMatriz” y “animación”:

Nombre	Propósito
obtenerMatriz	Este método consigue en la base de datos los créditos de cada estudiante de una cohorte, para todos los trimestres cursados por dichos estudiantes, para la generación de las instantáneas usadas en las animaciones.
animación	Genera una animación básica cuando no tiene especificados datos de entrada, y una instantánea correspondiente a una cohorte según los datos dados.

En la aplicación “multigraph” se usan las funciones “obtenerMatriz” y “multigrafica”:

Nombre	Propósito
obtenerMatriz	Para cada una de las cohortes ingresadas, este método consigue en la base de datos los créditos de cada estudiante de una cohorte, para todos los trimestres cursados por dichos estudiantes, para la generación de las instantáneas usadas en las animaciones.
multigrafica	Genera una animación básica cuando no tiene especificados datos de entrada, y de poseer datos, construye la animación que compara las cohortes solicitadas.

Capítulo 4

Trabajo en equipo

Para el desarrollo del proyecto se identificó las diferentes necesidades por cada historia de usuario. Se logra identificar las siguientes actividades: la animación del flujo de estudiantes de dos o más cohortes, cambio de la granularidad de los diagramas de barra, la generación de un gráfico de curvas que permita visualizar el flujo de estudiantes de una o más cohortes, la realización del informe. Se trabajó por pares en las diferentes actividades, dos parejas se encargaron de realizar la animación de varias cohortes, ya que esto requería más trabajo, una pareja se encargó de realizar las vistas e implementar los botones para cambiar la granularidad, la otra se encargó de generar el gráfico de curvas, finalmente el informe fue realizado por cada integrante del grupo.

La actividad que requirió más esfuerzo y tiempo fue la animación de varias cohortes que se realizó en aproximadamente 3 horas diarias durante 5 días. El cambio de granularidad tomó 1 hora diaria durante 3 días y el gráfico de barras se realizó en 8 horas. El informe requirió 3 horas.

Capítulo 5

Experiencia TDD

Para el último sprint se realizaron pruebas unitarias utilizando las técnicas de análisis de frontera y malicia. Para ello se creó un archivo `test.py`, en con la opción de cargar 5 cohortes, 5 trimestres en donde para cada cohorte se carga un estudiante con la cantidad de créditos que tiene aprobado y el respectivo trimestre. Con estos datos se procedió a realizar las diferentes pruebas para cada historia de usuario: Para la animación de la gráfica se realizó dos pruebas borde, una contenía la cantidad máxima de cohortes que se pueden mostrar que en nuestro caso es 5 y la cantidad mínima la cual es 1, dos pruebas malicias, una en donde se seleccionaba más de 5 cohortes y una en

donde no se seleccionaba ninguna. Para la granularidad también se realizaron dos pruebas bordes una que contenía granularidad igual a 5 y la otra de 240, y una prueba de malicia que consistía en no escoger ninguna granularidad. Respecto a la cohorte se realizaron pruebas en donde se escogía las cohorte 17 y en donde no se escogía ninguna. Con la velocidad también se realizó las pruebas en donde no se escogía una velocidad y otra en donde se escogía el máximo que es 3000 milisegundos.

Al trabajar con TDD pudimos comprobar que cada requerimiento que estábamos implementando estaba funcionando correctamente, ya que al realizar estas pruebas pudimos corregir detalles, los cuales hubieran sido imposible de ver si no se realizaban dichas pruebas.

Capítulo 6

Manejo del repositorio Github

Para el desarrollo de este sprint se creó una nueva rama llamada “desarrollo3”, al final del desarrollo del sprint se hizo merge con la rama “master”. Se turnaron los diferentes commits que se hicieron en la rama desarrollo. No se usaron más ramas ya que se realizó una historia de usuario a la vez y hasta que no se terminó no se procedió con la siguiente historia.

Capítulo 7

Conclusiones y recomendaciones

Con este tercer sprint se completó el primer prototipo funcional para presentar al cliente y decidir si se prosigue con el desarrollo del Avanzómetro para uso de la Universidad Simón Bolívar.

Se recomienda continuar con el desarrollo del Avanzómetro ya que puede ser de una gran utilidad para mejorar el desempeño de los estudiantes de la universidad, sin embargo es necesario realizar un análisis previo del propósito en concreto de la aplicación, y como llegar a ello. Recomendamos que se analicen los tipos de datos a colocar en la gráfica: si deben ser cantidad de estudiantes vs créditos aprobados, cantidad de reprobados vs trimestre, aprobados y reprobados con un conjunto de materias en inscritas en el mismo trimestre, o alguna otra comparación relacionada.

Además es recomendable realizar una investigación de la documentación de la librería d3.js de JavaScript ya que esta contiene una amplia gama de graficas que pueden ayudar a visualizar mejor los datos necesarios. También se recomienda estudiar en lenguaje de programación estadístico R, ya que podría ser de utilidad para el propósito de esta aplicación.