

Manual Técnico

Para

Nitrate

Versión 1.0 aprobada

Preparado por:

Josué Arrieta Salas

Adrián López Quesada

Seth Michael Stalley

20/1/2017

Historial de Revisiones

Nombre	Fecha	Descripción	Versión
Manual Técnico 1	20/1/2017	Creación del primer documento del manual técnico para el Sistema Nitrate. Acorde a todas las iteraciones y funcionalidades del sistema.	1.0

Índice

Requisitos	5
Visión General.....	5
Especificación revisada	6
Primera iteración (aplicación de escritorio).....	6
Segunda iteración (aplicación de escritorio).....	6
Tercera Iteración (aplicación móvil).....	7
Manual de usuario	7
Funcionamiento	8
Análisis del problema	10
Diseño	13
Visión General.....	13
Estructura en tiempo de ejecución.....	15
Estructura del módulo.....	24
Pruebas	28
Estrategia	28
Pruebas automáticas	28
Pruebas manuales	29
Resultados del test.....	30
Reflexión.....	31
Evaluación.....	31
Lecciones	32
Errores y limitaciones conocidos	32
Apéndice.....	33
Formato.....	33
Especificaciones del módulo	34
Casos de prueba	34
Subir archivo de texto (aplicación escritorio)	35
Leer archivo de texto (aplicación escritorio)	35
Generar la fórmula de concentración (aplicación escritorio)	36
Observar carpeta (aplicación escritorio)	37

Calcular el valor de “Sample” (aplicación escritorio).....	38
Ingresar el valor de concentración estándar (aplicación escritorio)	38
Guardar proyecto (aplicación escritorio).....	39
Abrir proyecto (aplicación escritorio)	39
Exportar a Excel (aplicación escritorio)	40
Generar gráfico concentración - tiempo (aplicación escritorio)	41
Generar gráfico absorbancia - concentración (aplicación escritorio)	42
Exportar gráfico a imagen (aplicación escritorio).....	43
Generar gráfico concentración vs. tiempo (móvil)	44
Generar gráfico absorbancia vs. concentración (móvil).....	45
Mostrar valor de concentración (móvil)	45
Exportar gráfico a imagen (móvil)	46
Glosario.....	47

Requisitos

La sección de requisitos describe el problema que se está solventando junto con la solución. Esta sección será tanto para los usuarios como para los implementadores.

Visión General

La funcionalidad primordial u objetivo principal del sistema a implementar es calcular la concentración de Nitratos a partir de un espectrómetro en una muestra de agua dada, y de ahí recae la importancia de este: automatizar el proceso mencionado para aumentar la eficiencia de los cálculos; y de esta manera facilitar el trabajo de la profesora Laura. Calcular esta concentración de nitratos es importante, ya que si pasa de cierto rango de número, el agua no se considera potable y podría perjudicial para la salud humana si se consume. También la aplicación debe mostrar los cálculos obtenidos visualmente, poder establecer relaciones entre estos y lograr predicciones a futuro.

Para cumplir con el objetivo principal recién mencionado se piensa desarrollar un software que será multi-parte para cuantificar la concentración de nitratos a partir de una muestra de agua; y mostrar los resultados visualmente utilizando tablas y gráficos. Habrán dos sistemas: uno para un computador de escritorio y otro que funcionará en un celular inteligente. Tales mediciones de parámetros se consiguen a partir de un espectrómetro. Se buscará que el sistema pueda exportar estos resultados de diferentes maneras, como por ejemplo un archivo de imagen o una tabla Excel. También podrá realizar correlaciones entre los datos obtenidos de las muestras de agua y empezar a realizar predicciones de estos parámetros.

Es importante mencionar que por medio del sistema móvil, se podrá ver dicha información en tiempo real. Se podrán comparar datos y crear nuevas calibraciones desde el sistema de escritorio.

Actualmente la profesora Laura Hernández cuantifica la concentración de nitratos de muestras de agua con un espectrómetro de forma manual. El sistema Nitrate vendrá a solventar este problema y automatizar todo el proceso; así como para la visualización de resultados en tiempo real. Es importante mencionar que debido a la naturaleza tan específica del proyecto para la cual el sistema se está creando, no existe otro sistema que cumple con todos los requerimientos requeridos por el usuario.

Este proyecto será realizado en el curso Proyecto de Ingeniería de Software de la Escuela en Ingeniería en Computación del Instituto Tecnológico de Costa Rica a cargo de la profesora Ing. María Estrada Sánchez. Es importante mencionar que este proyecto surge a necesidad de la profesora Laura Hernández de la Escuela de Química.

Especificación revisada

Para la realización de este documento se tomará la funcionalidad total del Sistema Nitrate de la versión 1.0. Tal funcionalidad esencial se puede encontrar en el listado de casos de uso (<https://drive.google.com/open?id=0Bwn9E8E9d8OwZUEtNURlaFkxZU0>), donde se comprenden los 16 casos de usos primordiales, divididos por iteraciones:

Primera iteración (aplicación de escritorio)

Identificador	Caso de uso	Prioridad
UC-001	Subir archivo de texto	Alta
UC-002	Leer archivo de texto	Alta
UC-003	Generar la fórmula de concentración	Alta
UC-004	Observar carpeta	Media
UC-005	Calcular el valor de “ <i>Sample</i> ”	Media
UC-006	Ingresar valor de concentración estándar	Baja

Segunda iteración (aplicación de escritorio)

Identificador	Caso de uso	Prioridad
UC-007	Guardar proyecto	Alta
UC-008	Abrir proyecto	Alta
UC-009	Exportar a Excel	Media
UC-010	Generar gráfico concentración - tiempo	Media
UC-011	Generar gráfico absorbancia - concentración	Media
UC-012	Exportar gráfico a imagen	Baja

Tercera Iteración (aplicación móvil)

Identificador	Caso de uso	Prioridad
UC-013	Generar gráfico concentración vs tiempo	Alta
UC-014	Generar gráfico absorbancia vs concentración	Media
UC-015	Mostrar valor de concentración	Media
UC-016	Exportar gráfico a imagen	Baja

Es importante mencionar que las especificaciones detalladas del comportamiento del sistema fueron especificadas y aclaradas mediante reuniones tanto presenciales como virtuales con Laura Hernández. De tal manera que no es necesario aclarar cualquier suposición o que se haya hecho sobre el significado de los requerimientos o casos de usos de lo mismos. No hay ambigüedad en los requerimientos. No fue necesaria la modificación de ninguno. Sin embargo si hubo extensión de requerimientos, en los que se implementaron dos casos de uso considerados como extra (también se pueden encontrar en el documento de listado de casos de uso). Estos no forman parte de las funcionalidades esenciales del sistema:

1. Poder generar cualquier gráfico mencionado anteriormente de forma manual con base a un conjunto de archivos seleccionados.
2. Poder configurar los distintos gráficos de acuerdo a un valor ingresado, establecer un color para el gráfico. *Alert Values*.

Manual de usuario

Para el uso detallado de cómo se debe utilizar el sistema, se recomienda visualizar el documento del manual de usuarios para la versión correspondiente del sistema MOLABS. Este puede ser accedido por el siguiente link. Tal documento es muy extenso y por esta razón se considera pertinente hacer una referencia:

<https://drive.google.com/open?id=0BzER2fgK5TZ-VUVJM202SGlocFE>.

Sin embargo, a continuación se hará una breve explicación de ciertos factores que todos los usuarios de sistema (*owners*) deben saber.

El sistema solo puede ser accedido por quien contenga una cuenta registrada en el mismo. De esta manera solo las personas autorizadas pueden tener acceso a este, algunos con menos privilegios que otros. Los usuarios de sistema (*owners*) tienen todos los privilegios, estos pueden eliminar usuarios, crear cualquier tipo de usuarios y hacer total manipulación del sistema y su ambiente de trabajo. Posteriormente los usuarios de tipo administrador solo pueden crear usuarios de tipo *user* los cuales tienen la menor cantidad de privilegios, sin embargo los administradores gozan de los mismos privilegios para manipular el ambiente de trabajo del software como los de sistema. Finalmente, los usuarios normales o *users* tienen la restricción de que solo pueden usar el software para visualizar algún proyecto que les fue enviado por algún administrador u *owner*. Estos no pueden crear usuarios ni manipular ningún tipo de dato.

Otro aspecto que deberían conocer los usuarios de sistema, es que estos pueden manipular cualquier otro usuario, cambiar sus credenciales o información y modificar su tipo, lo que implica quitarles o darles más privilegios. Sin embargo, un usuario sistema no puede involucrarse con otro de sistema por razones de seguridad. Además, cualquier usuario puede utilizar la aplicación móvil/web sin ningún problema, mientras tenga correctas sus credenciales, ya que esta solo sirve para visualización y no hay ningún tipo de modificación de datos. Finalmente, los archivos que contengan un proyecto de MOLABS están encriptados, y solo pueden ser descryptados por el programa la hacer conexión con un usuario cualquiera, por lo tanto no hay problema de perder o divulgar información.

Funcionamiento

Para que el sistema de escritorio pueda operar de manera correcta, se necesita que este tenga al menos las siguientes características:

- Computador con java 8 en adelante instalado.
- Contar con un computador con una versión reciente del sistema operativo de Windows o Mac OS.
- 512mb de RAM.
- 20mb de disco duro.
- Procesador Intel i3 o superior.

Para que el sistema de celular pueda operar de manera correcta, se necesita que este tenga al menos las siguientes características:

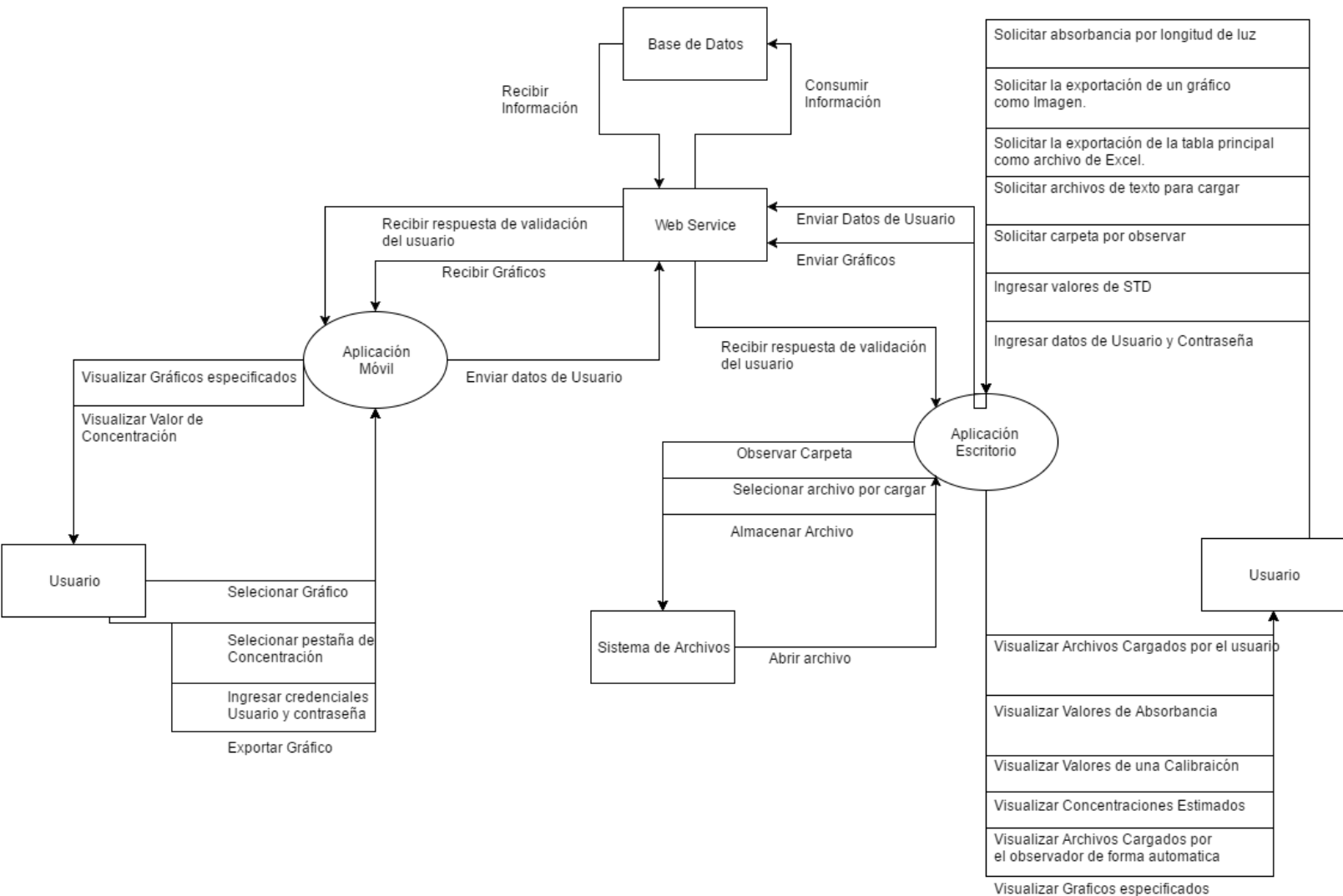
- Celular inteligente con alguna versión inteligente de iOS (10.0 en adelante) o Android (6.0 en adelante).
- Dispositivo móvil con un navegador web capaz de correr JavaScript y HTML 5 (no se puede utilizar navegadores mini, como Opera).
- 10 Mb de disco duro como máximo.
- 64 Mb de RAM.

Para utilizar ambas versiones se recomienda tener una conexión a Internet estable. También para ambos sistemas, se espera que se consuman los siguientes recursos como mínimo (tomado de los requerimientos de rendimiento del Estatuto de Requerimientos del sistema <https://drive.google.com/open?id=0Bwn9E8E9d8OwQXZIWmlUMVM5dDA>):

REQ-089	El sistema debe utilizar un máximo del 10% del total de memoria principal del dispositivo.
REQ-090	El sistema para cada funcionalidad o característica que posea, se espera que la realice en un máximo de 3 segundos.

Análisis del problema

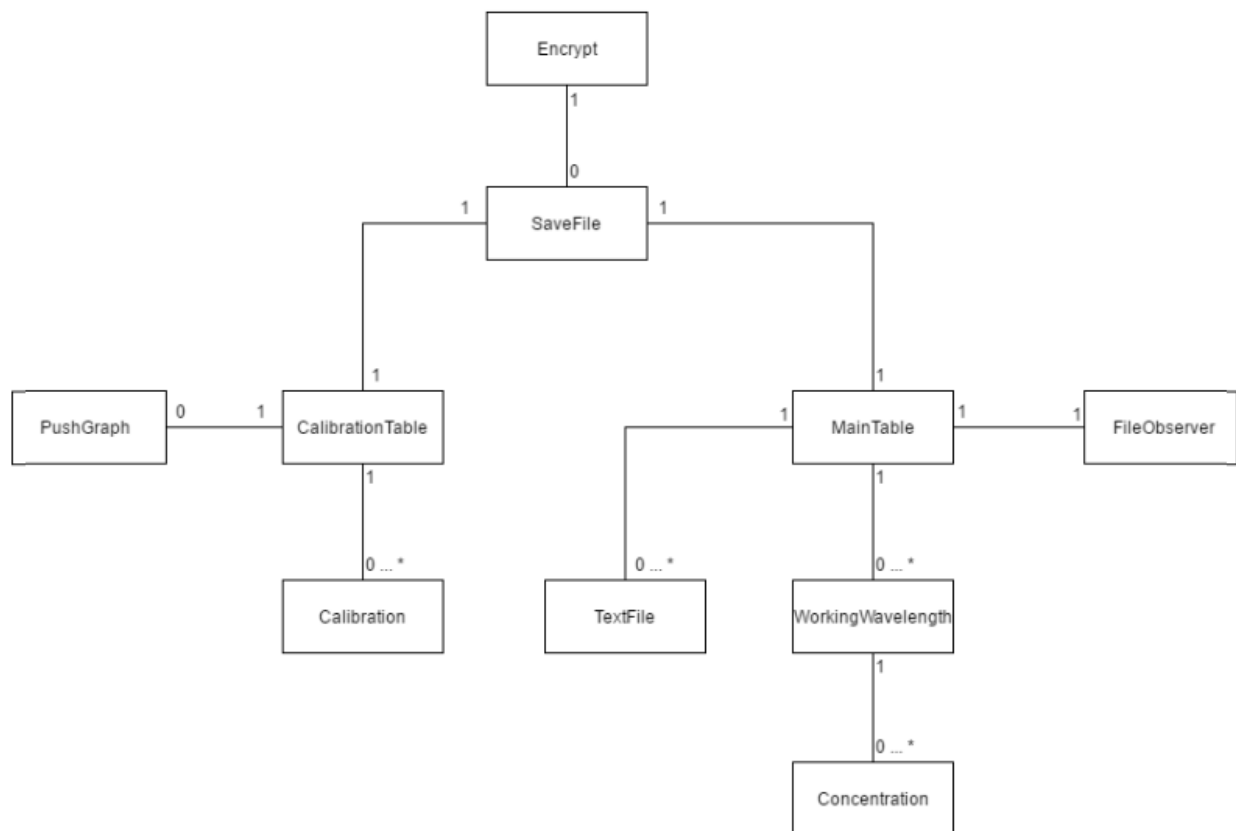
En primer lugar para mostrar el sistema Nitrate de manera general y como un todo, se utilizará un diagrama de contexto:



Se puede ver que la aplicación móvil y la aplicación de escritorio se conectarán al mismo servicio web. Este servicio web será implementado utilizando NodeJS en JavaScript y estará en la nube de Amazon. La primera corresponde a funcionalidades de visualización y exportación de datos. La segunda corresponde a las mismas funcionalidades, pero también en la generación de datos. El servicio web estará conectado con un motor de base de datos relacional en MySQL 5.6. El contexto de dominio se puede observar de manera clara en tal gráfico.

Para mostrar la lógica de la aplicación de escritorio y la aplicación de celular de manera general, se utilizará un diagrama de dominio y otro de componentes. Este con el propósito de realizar un análisis del problema y proveer una solución a nivel de diseño. Es importante mencionar que no hay asuntos que quedaron pendientes o que no se hayan resueltos, se presenta la totalidad del sistema en este documento.

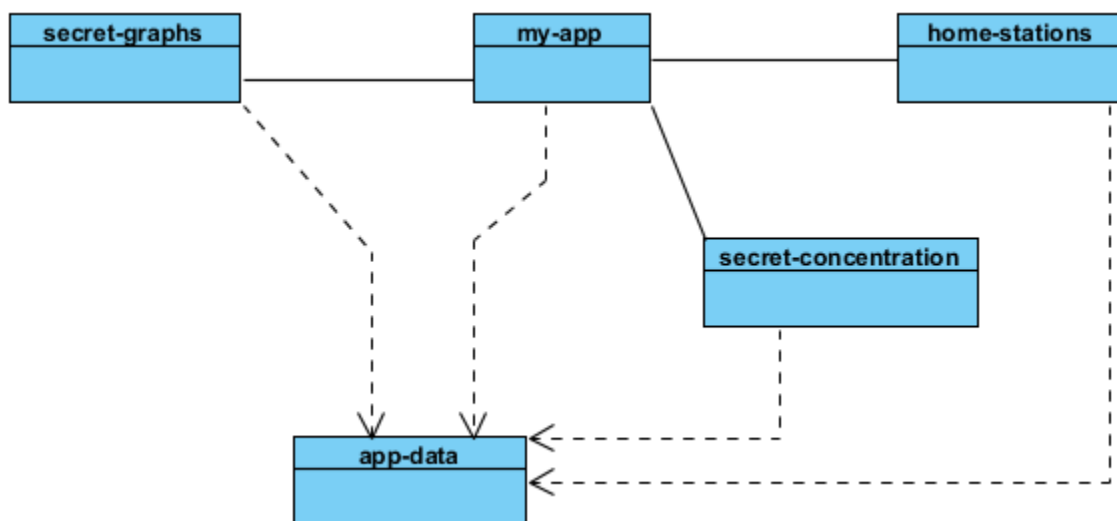
Para la elaboración del proyecto Nitrate para la versión de escritorio se tiene el siguiente modelo de dominio:



Este modelo de dominio es una vista abstracta de las relaciones entre los componentes principales del modelo. Primero, tenemos que los dos componentes principales del sistema que son: “CalibrationTable” y “MainTable”, que son la tabla de calibraciones y la tabla principal del sistema. La tabla de calibraciones opera con cero o más calibraciones, los cuales son manejados por dicha clase. Segundo, la tabla principal tiene un observador de archivos y dicha se relaciona con cero y más archivos (generados por un dispositivo de hardware) y cero o más “WorkingWavelengths” que son las absorbancias que el usuario está actualmente utilizando; además, estas absorbancias tienen a cero más concentraciones que son calculados a partir de

dicha absorbancia y la fórmula de calibración. Finalmente, existe un “SaveFile” el cual refleja el estado completo de todos los componentes relacionados a ella, el cual es utilizado para almacenar y recuperar el modelo. La entidad *PushGraphs* será la encargada de consumir el servicio web de manera constante para enviar los gráficos de un usuario dado; con el propósito de que se puedan observar desde la aplicación para teléfono móvil.

Para la aplicación de celular no se tiene un modelo de dominio ya que no hay lógica de negocios involucrada. También por la tecnología utilizada (*webapp* utilizando *Polymer*), no se puede generar un diagrama UML de dominio. No hay manipulación de datos y estos solo son mostrados en pantalla en forma de gráficos y texto. Lo que hay es un manejo de vistas. Sin embargo se puede crear un diagrama de modelo de componentes de la siguiente forma para explicar su funcionamiento. Se presenta en forma de diferentes vistas cada una con su propia responsabilidad:



- *my-app*: sería el punto de entrada del programa. Maneja las referencias entre las páginas.
- *home-stations*: funciona como la página de entrada del programa. Representa el *login* y la pantalla con las estaciones activas de un determinado momento.
- *secret-graphs*: funciona como vista para los gráficos: concentración vs tiempo y absorbancia vs concentración (junto con sus datos de calibración).
- *secret-concentration*: será la vista dónde se mostrará el último valor de concentración en

grande de forma descriptiva de acuerdo a los alert values.

- *app-data*: es el manejador de *localStorage*, de manera que cuando un dato de *localStorage* cambia esta notifica a los diferentes suscriptores: *secret-graphs*, *secret-concentrations*, *home-stations* y *my-app*. De esta manera los datos mostrados cambian.

Diseño

La sección de diseño de su documentación proporciona un cuadro de alto nivel de su estrategia de implementación

Visión General

Para comprender el diseño, en primer lugar se deben considerar las herramientas que se utilizaron para la realización del proyecto:

- El motor de base de datos relacional MySQL 5.6.
- Webservice en NodeJS 6.9.4 para la implementación del servicio web. Estará hosteado en un servidor de Amazon. Será intermediario entre la base de datos y tanto el sistema móvil como el de escritorio. Todo dato enviado hacia dentro o hacia fuera del servidor deberá estar en formato JSON. La conexión con este para la aplicación de escritorio será por medio del protocolo https para la aplicación móvil será http.
- Para la aplicación móvil se utilizará como lenguaje de programación Java 8 con el ambiente de desarrollo Java Eclipse Neon.
- Para la aplicación móvil se desarrollará en los lenguajes habituales de un ambiente web: JavaScript, CSS y HTML.
- Para la realización del prototipo de ambos sistemas se utilizó Justinmid versión 6.7.0

Para la realización específica de la aplicación de escritorio se utilizarán las siguientes librerías:

- Gson-2.8.0.jar: se utilizará para la serialización de objetos en Java a JSON y viceversa.
- Commons-math3-3.2.jar: se utilizará para la realización de distintos cálculos: intersección de una recta con el eje y, su pendiente. Calcular índice de *Pearson* y correlaciones entre variables.

- Junit.jar: la versión 4. Se utilizará para realizar todo tipo de pruebas: de sistema, unidad e integración.
- HttpClient-4.5.2.jar: se utilizará para realizar conexión con el servidor web y poder realizar consultas http de tipo post.
- json20090211.jar: se utilizará para el manejo de archivos json que van y vienen del servidor web.
- jxl.jar: se utilizará para la integración con Excel. De esta manera se podrá exportar la tabla principal a un archivo Excel.
- gral-core-0.11.jar: se utilizará para la generación y exportación de gráficos en la aplicación de escritorio.

Para la realización específica de la aplicación móvil se cuentan con las siguientes librerías o frameworks:

- Polymer 1.6: es una librería que se utilizó como base de toda la aplicación móvil. Extiende y facilita la manipulación de la página.
- highcharts 2.0.0: es una librería se utilizó para la generación, explotación y manipulación para los gráficos de la aplicación móvil.

No se considera que hubo decisiones tomadas que puedan haber resultado erróneas. Se considera que fue un proceso directo, analizado y organizó a pesar de la limitante de tiempo que se tenía. Se aclaró siempre con Laura Hernández y María Estrada cualquier aspecto que pudo haber quedado ambiguo; que haya podido ser la causa de decisiones de diseño erróneas. Fue un proceso continuo, incremental e iterativo.

Se ha de mencionar que el pilar fundamental de la estructura del diseño para el sistema Nitrate para la versión de escritorio fue el patrón de diseño: modelo-vista-controlador (MVC). Este consiste separar el sistema en al menos tres capas: paquetes de entidad, paquetes de interfaz y paquetes de controlador. También se crearon paquetes transversales de validación y acceso a la base de datos. Este patrón nos permite un sistema altamente reusable y un diseño modular altamente mantenible.

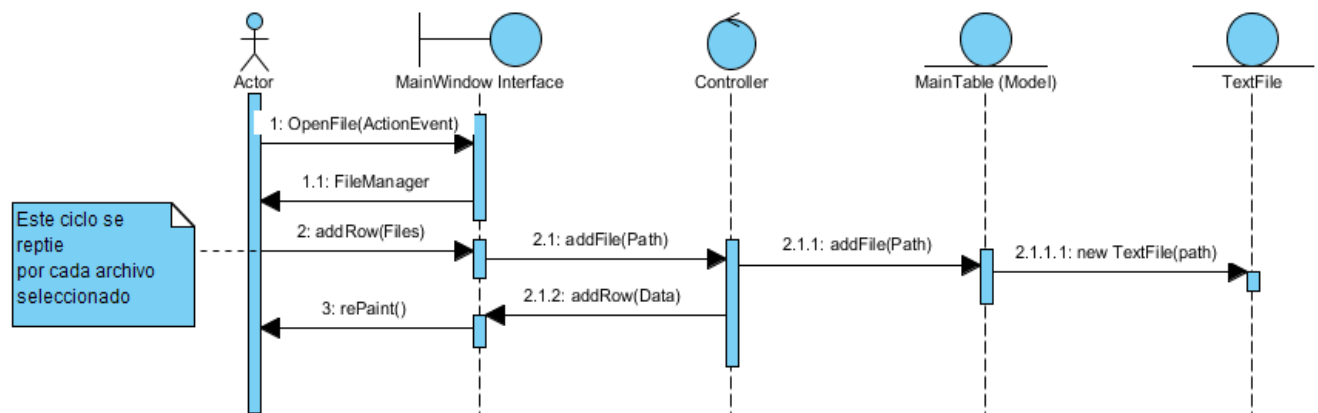
Un aspecto clave en el diseño es cómo se guardaran los distintos gráficos de la aplicación de forma persistente. Esto es en la base de datos. Se optó por almacenar cada gráfico como un

archivo con formato JSON. De tal manera que los puntos 'x' y 'y' se guarden como una lista ordenada dentro del archivo. También se pueden incluir otros datos como: *Pearson*, *wavelength*, intersección con el eje y, y la pendiente.

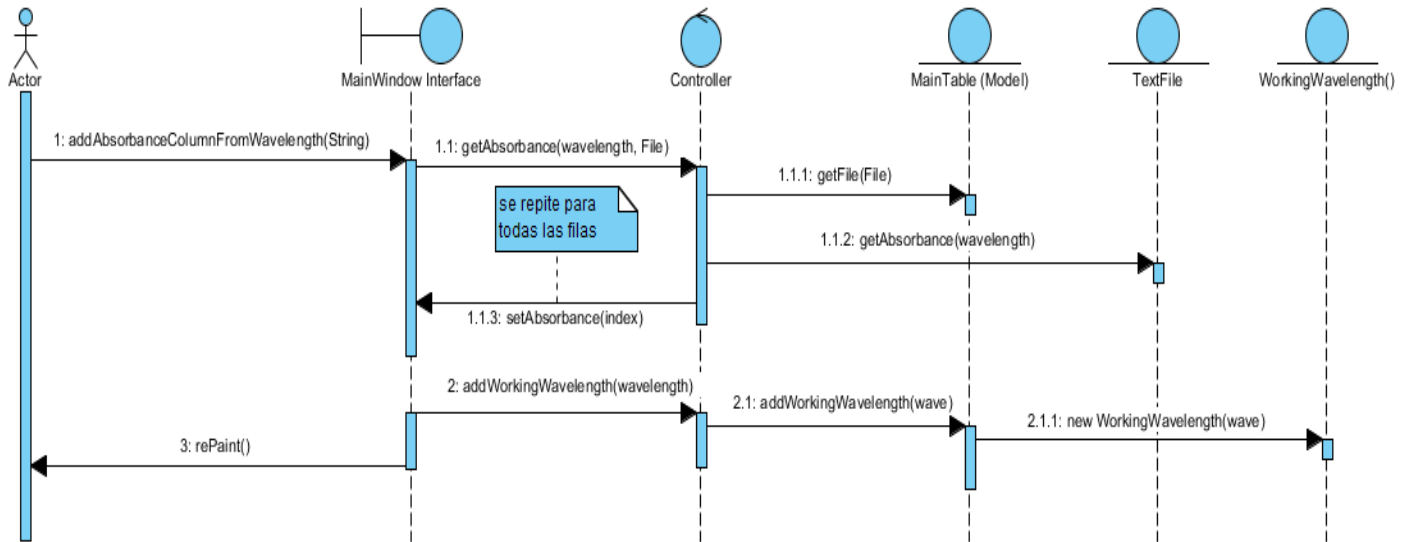
Estructura en tiempo de ejecución

Esta sección se describe la estructura del estado del programa en ejecución. En otras palabras amarrar la estructura con el comportamiento del sistema. Se describirán cada caso de uso por medio de un diagrama de secuencia el cual contendrá las funciones e interacciones principales del sistema, enfocadas en la parte de la lógica de negocios. No se hará mucho enfoque a la interacción de los componentes gráficos ya que esto haría los diagramas más extensos, y se quiere hacer énfasis en los componentes lógicos. En todos los diagramas se tomarán en cuenta los pasos de cada caso de uso que generalmente asumen que el usuario ya está ingresado dentro del sistema o ya tiene datos dentro del sistema.

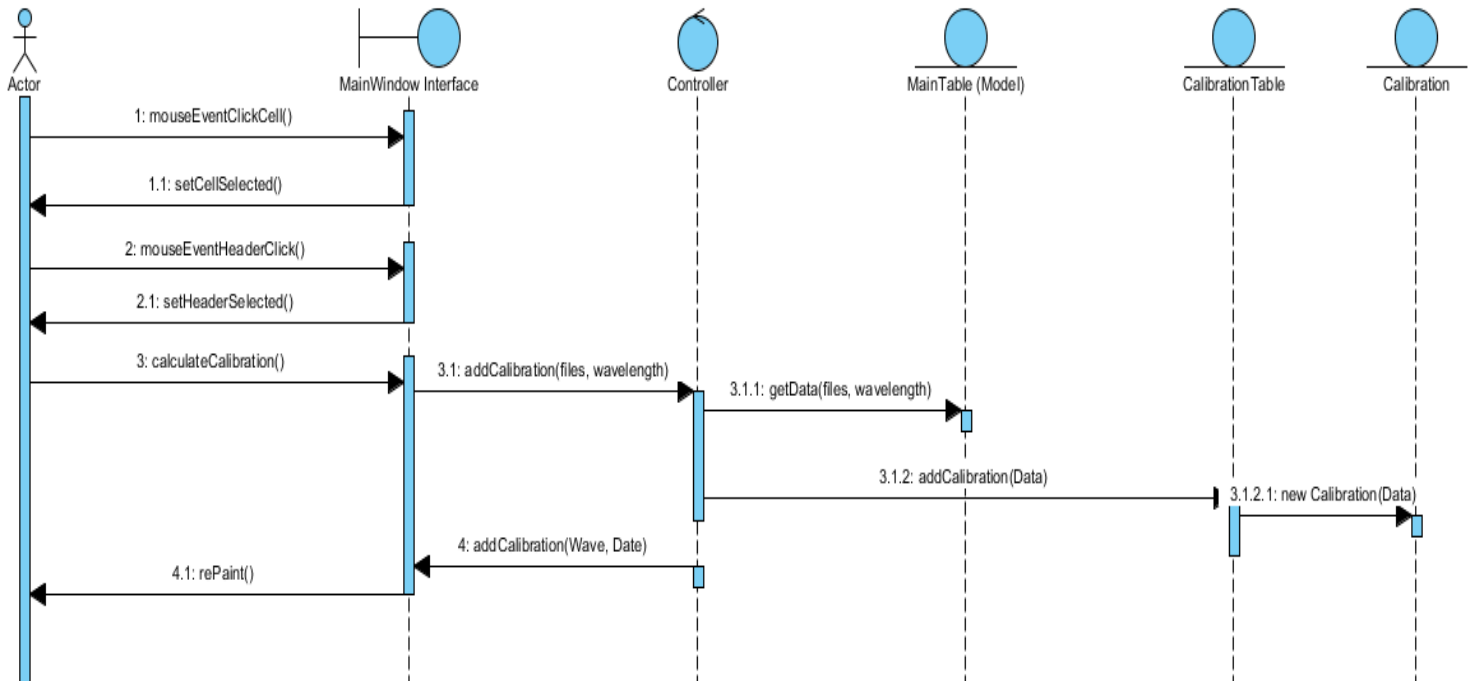
UC-001 - Subir archivo de texto (aplicación de escritorio)



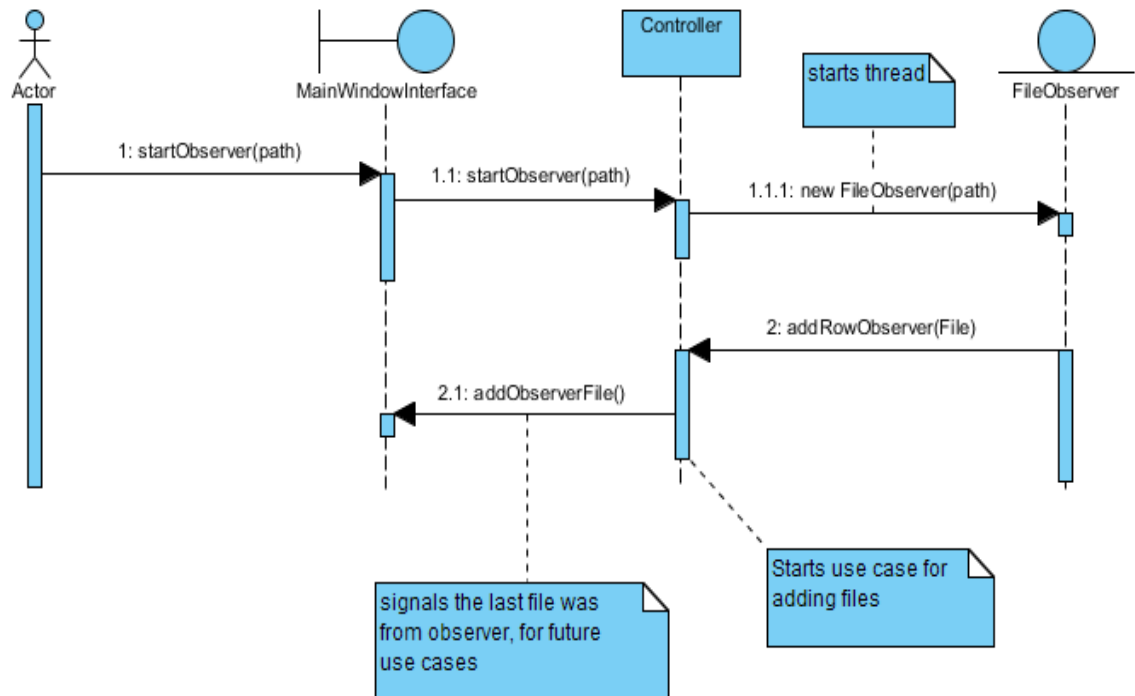
UC-002 - Leer archivo de texto (aplicación de escritorio)



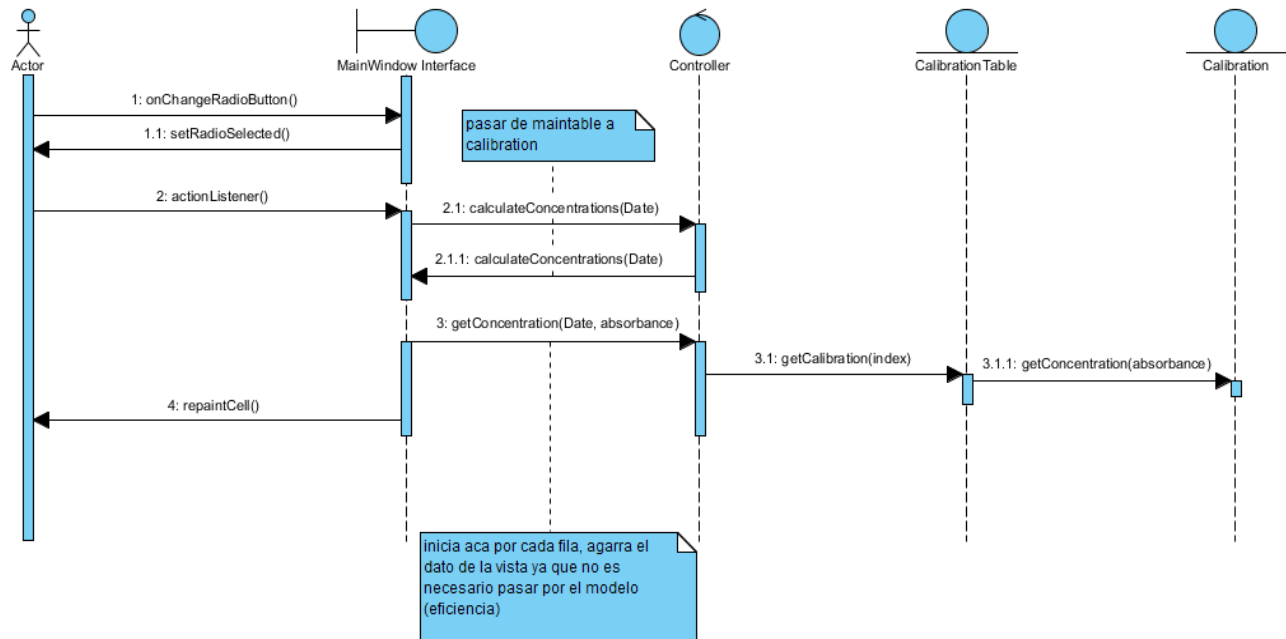
UC-003 - Generar la fórmula de concentración (aplicación de escritorio)



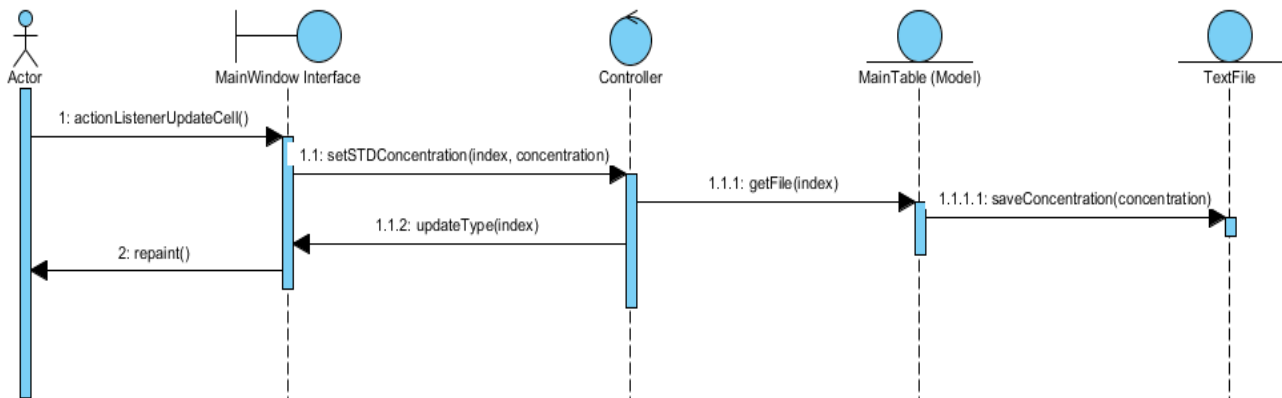
UC-004 - Observar carpeta (aplicación de escritorio)



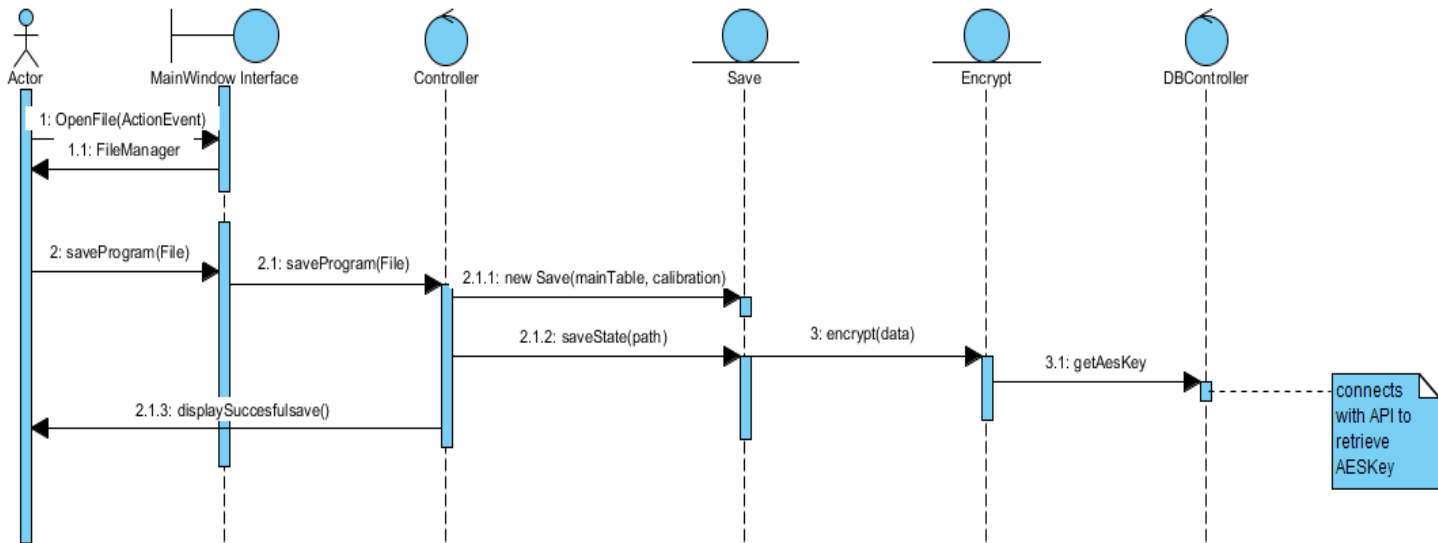
UC-005 - Calcular el valor de "Sample" (aplicación de escritorio)



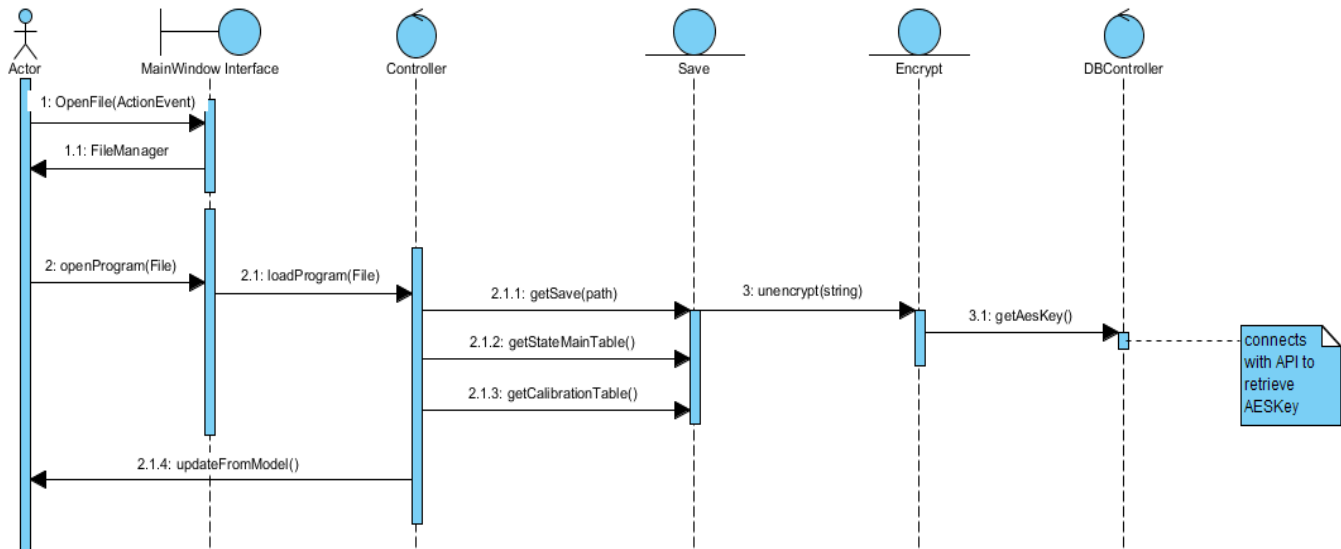
UC-006 - Ingresar el valor de concentración estándar



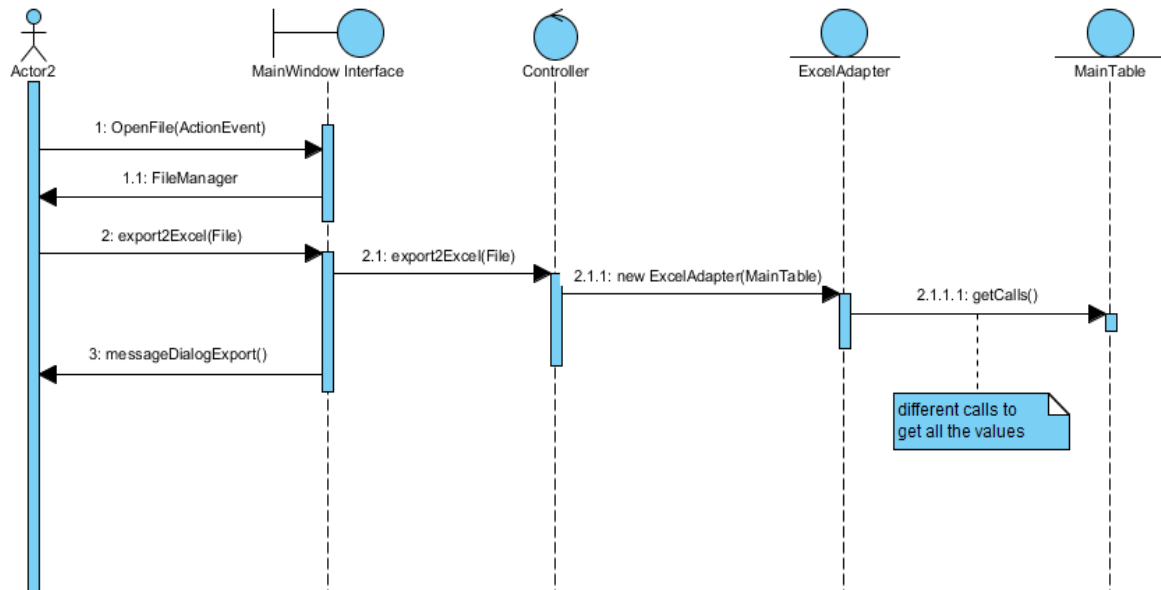
UC-007 - Guardar proyecto



UC-008 - Abrir proyecto

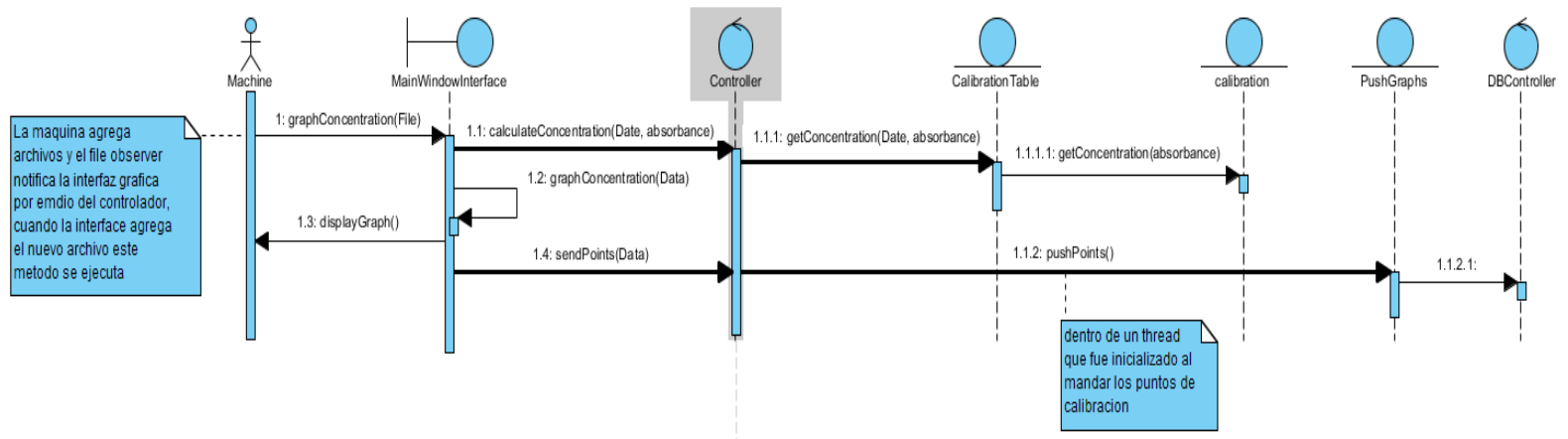


UC-009 - Exportar a Excel

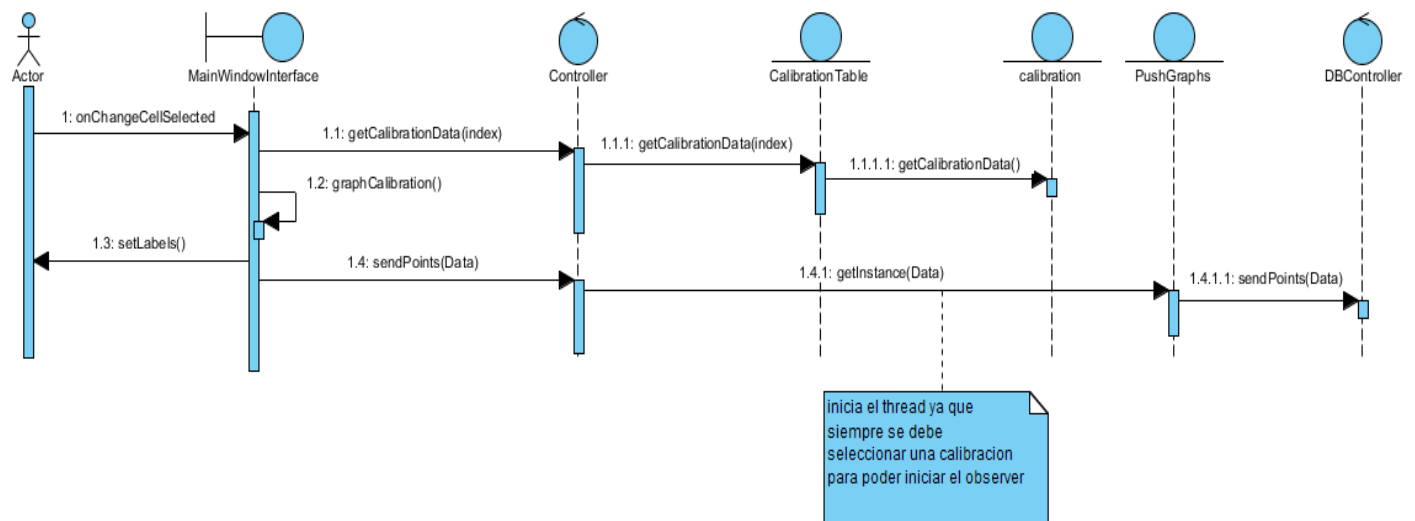


UC-010 - Generar gráfico concentración - tiempo

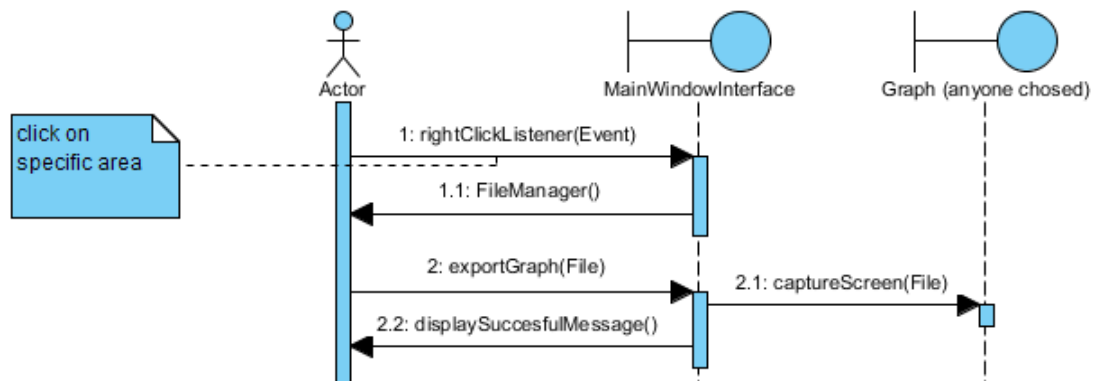
Para este modelo se parte desde el modelo del caso de uso de observar carpeta, ya que por cambios con el cliente se debe partir desde este punto.



UC-011 - Generar gráfico absorbancia - concentración

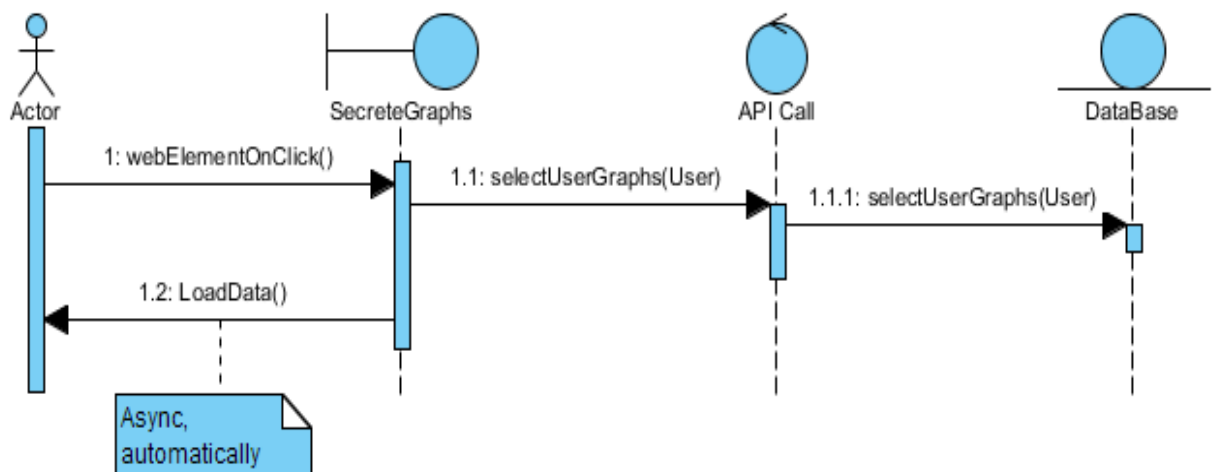


UC.012 - Exportar gráfico a imagen

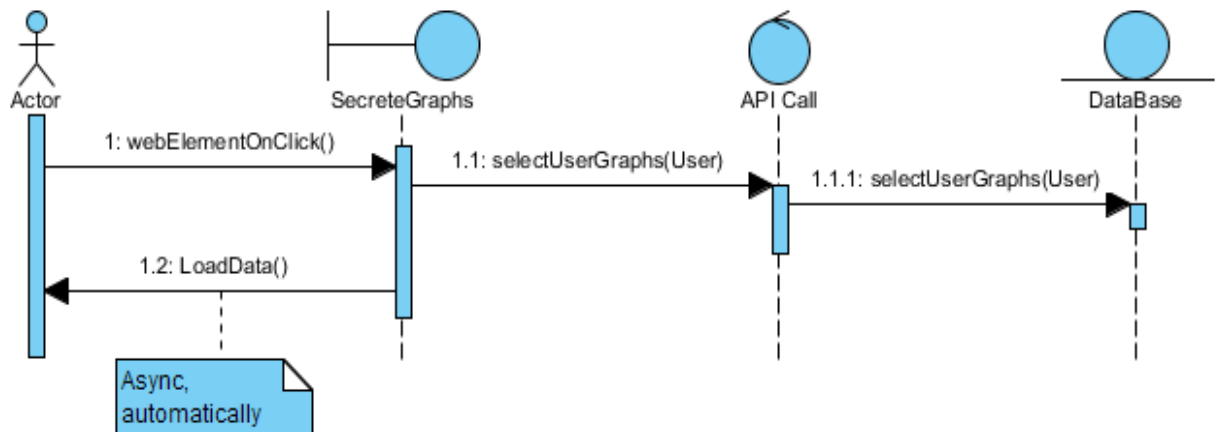


Los siguientes casos de uso parten desde la perspectiva que el usuario ya está ingresado, hay conexión a internet estable y ya se ha seleccionado una estación activa. Los siguientes dos son iguales ya que utilizan el mismo método y carga del gráfico. Corresponden a la aplicación móvil.

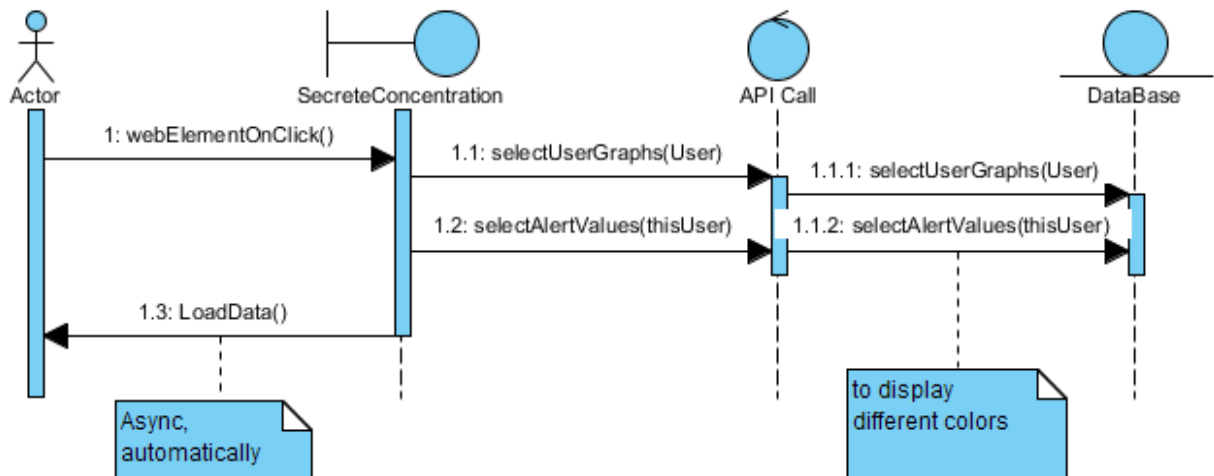
UC-013 - Generar gráfico concentración vs tiempo (Móvil)



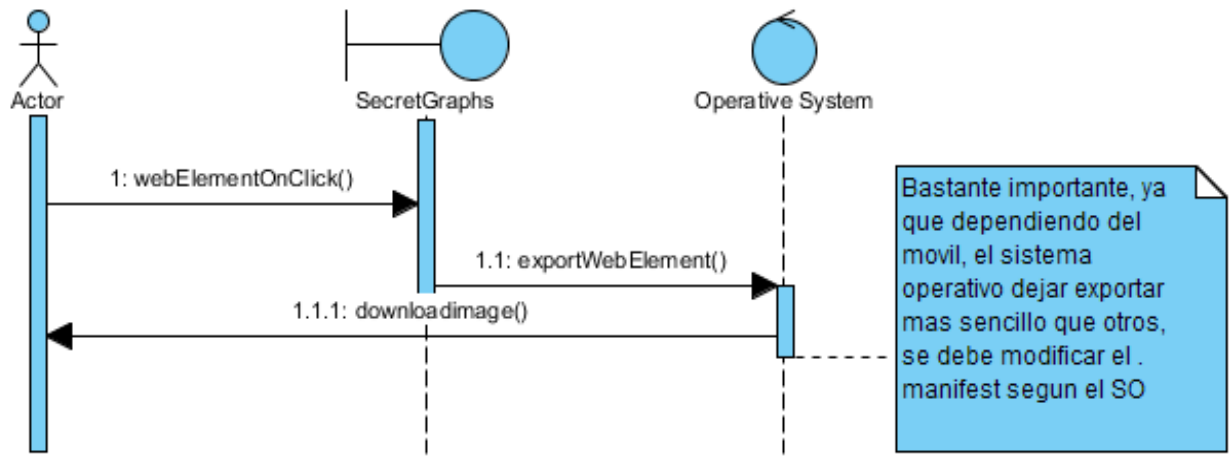
UC-014 - Generar gráfico absorbancia vs concentración (Móvil)



UC-015 - Mostrar valor de concentración (Móvil)

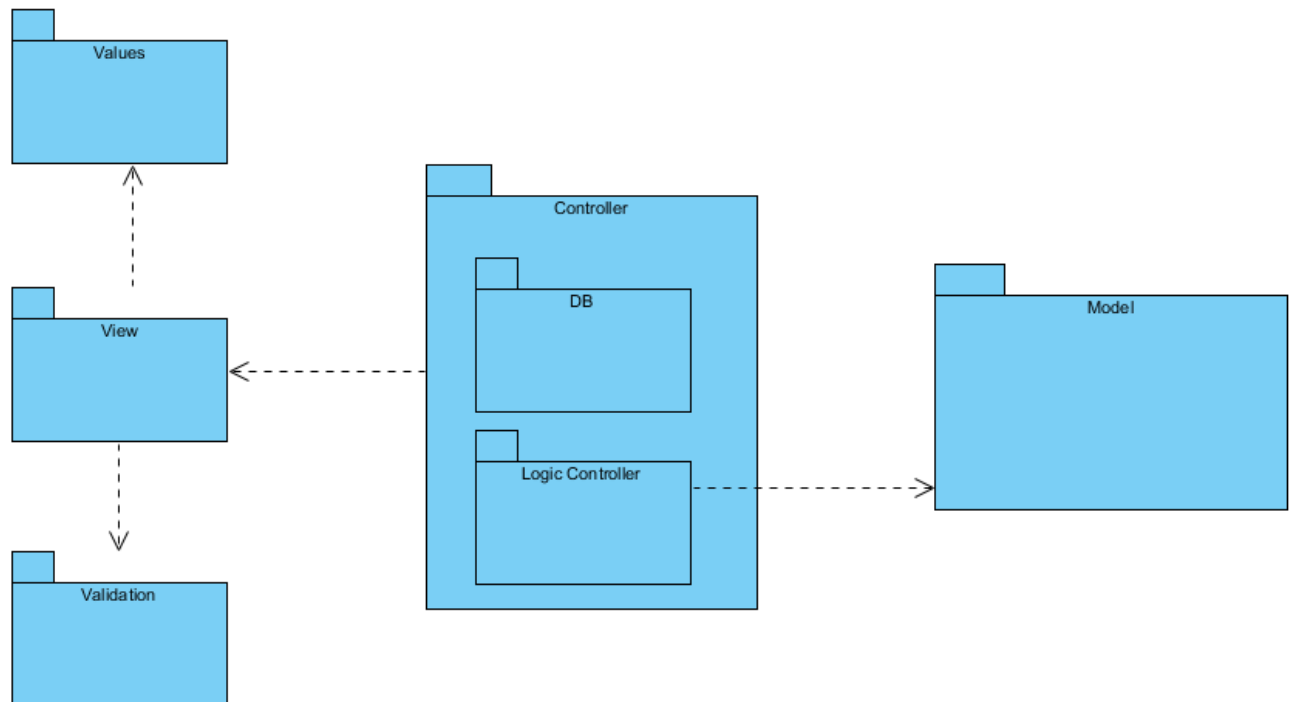


UC-016 Exportar gráfico a Imagen (Móvil)



Estructura del módulo

Para esta subsección se mostrará como los paquetes o módulos están organizados y cómo uno depende de otro. Se puede notar como se utilizó un modelo de vista y controlador como patrón de diseño principal para producir un código altamente reusable y fácil de mantener. También permite que el proyecto sea fácil de extender y la división de responsabilidades dentro del equipo de trabajo:



Los anteriores son los principales paquetes de la aplicación. Se enlistan:

- *Values*: es un paquete con clases de constantes utilizadas para los mensajes de error y configuraciones especiales de la interfaz gráfica. También el modelo lo utiliza para ciertas constantes.
- *Validation*: es un paquete especializado para la validación de entrada de datos del usuario. Esta validación se hará desde la interfaz gráfica.
- *View*: es un paquete que contiene todas las clases referentes a interfaz gráfica del sistema. Será el medio por donde el usuario interactuará con el sistema y podrá ingresar los distintos archivos de entrada.
- *Controller*: es un paquete que será la unión o el medio en que la interfaz podrá interactuar

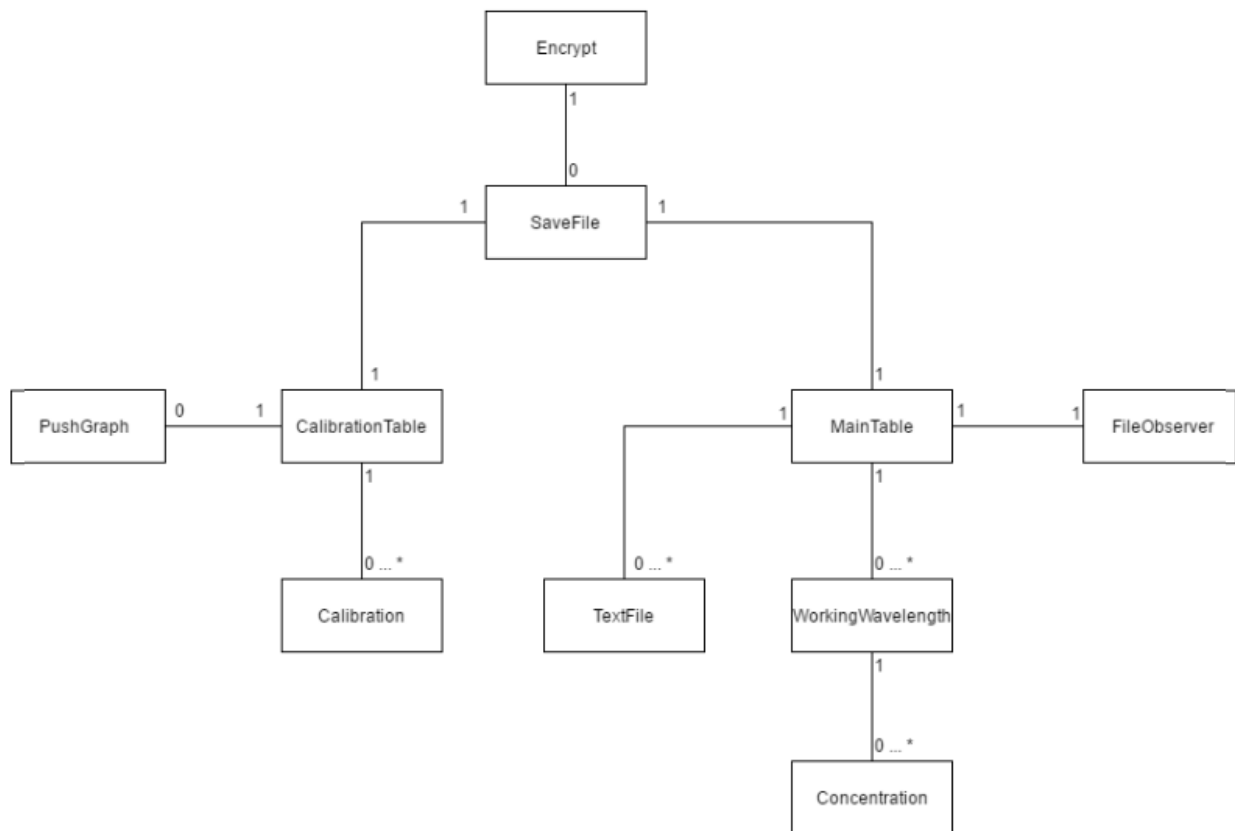
con la interfaz. De esta manera para mejorar la reutilización de paquetes y que sean independientes. Dentro de este paquete hay dos sub-paquetes: 1) DB: se encarga de controlar el acceso a la base de datos y se conectará al servidor web por medio del protocolo http. 2) LogicController este paquete será el controlador que accederá a la lógica de negocios.

- *Model*: este paquete contendrá toda la lógica de negocios. Realizará todo tipo de cálculos y estimaciones.

Se hablará de los paquetes solamente significativos y además las clases que estos poseen que se consideran importantes:

Model

Dentro del modelo se pueden enlistar las siguientes clases en el siguiente diagrama de dominio (el mismo que se mostró anteriormente):

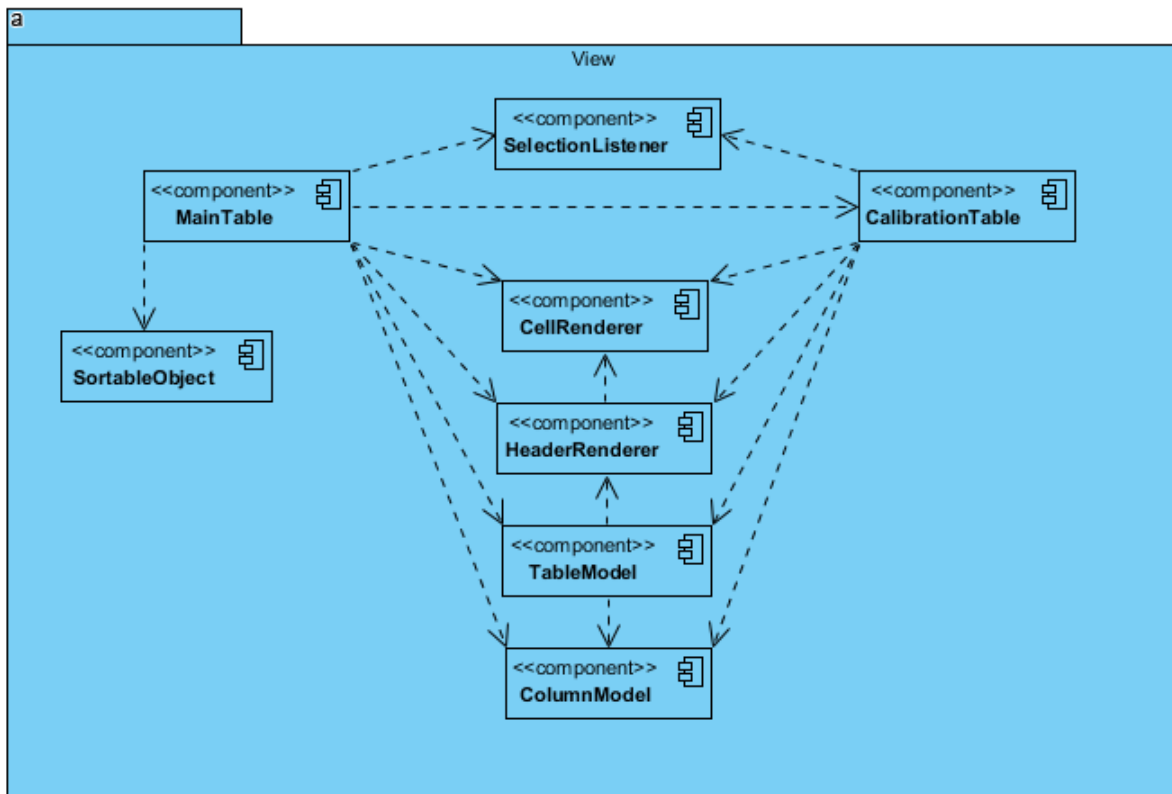


Se mencionan las clases más importantes:

- *MainTable*: se considera la clase más importantes de todas, y es una representación lógica de la tabla principal de la interfaz gráfica. Tendrá que siempre estar igual a lo que se ve en pantalla. En este estarán todos los archivos, las longitudes de onda para todos los archivos y la concentraciones calculadas a partir de esa longitud de onda. En la segunda iteración se tendrá que poder guardar el estado del proyecto, por lo cual guardar toda esta información en esta clase es imprescindible. Esta clase puede crear, modificar y eliminar tanto: absorbancias, archivos de texto y concentraciones.
- *CalibrationTable*: tiene el mismo funcionamiento que *MainTable*, excepto que será para la tabla de calibración. Deberá tener el mismo contenido que la interfaz gráfica ya que con esta se podrá cargar el estado de un proyecto. Esta clase podrá modificar, eliminar y crear nuevas calibraciones.
- *FileObserver*: esta clase es un thread encargado de observar la carpeta seleccionada de manera que todo archivo que ingrese a esta carpeta es cargado a *MainTable* y todas las diferentes absorbancias y concentraciones anteriores se calculen para este.
- *PushGraphs*: también esta clase será un thread que será la encargada de constantemente subir los gráficos de un determinado usuario a la base de datos. Se considera de suma importancia ya que sin esta clase en la aplicación de celular no se podría visualizar ningún dato ni gráfico. Se puede notar que hay una alta dependencia de la aplicación de celular con la aplicación de escritorio.
- *SaveFile* y *Encrypt*: estas se encargaran de guardar el estado completo de un proyecto en el sistema Nitrate y que se está se encripte con un algoritmo versión estándar del “Rijndael Cipher” AES.

View

Se tiene un diagrama de componentes de las clases consideradas como importantes:



Se quieren rescatar las siguientes dos clases:

- **MainTable**: es la tabla que el usuario ve y siempre deberá estar igual al modelo con su clase respectiva. Esta deberá mostrar (y tener toda interfaz para realizar) archivos, concentraciones y absorbancias. También hay un gran número de detalles de usabilidad que esta posee tales como: copiado y pegado de objetos, resaltamiento de columnas, orden de archivos cronológico, cambios de tipos automáticos, alargamiento si se desea de columnas, las concentraciones son insertadas a la par de su absorbancia, etc. Posee alto nivel de complejidad y se considera que la gran complejidad del proyecto está en esta clase y cómo representarla y actualizarla en el modelo lógico.
- **CalibrationTable**: es la tabla de calibración que el usuario verá en pantalla y deberá estar igual al modelo lógico con su clase respectiva. Deberá mostrar (y tener toda interfaz para realizar) calibraciones a partir de absorbancias dadas. También posee alto número de características de usabilidad tales como: cuando se selecciona una fila se muestran

resaltadas las absorbancias con la que se forma, copiado y pegado de objetos, etc.

Debido a que el sistema móvil corresponde a una arquitectura totalmente distinta, se tiene otro diagrama de paquetes. Sin embargo por la tecnología utilizada (*webapp* utilizando *Polymer*), no existen paquetes dentro del sistema, ni agrupaciones lógicas. Sin embargo se puede presentar el diseño de esta por medio de un diagrama de componentes, donde cada componente corresponderá a una vista con una responsabilidad única. Tal diagrama con su descripción se puede encontrar en este mismo documento en la sección de: Análisis del problema.

Pruebas

La sección de pruebas indica el enfoque que se ha elegido para verificar y validar su sistema. Para esta sección se recomienda altamente revisar el documento de plan de pruebas en su última versión que se puede encontrar en: https://drive.google.com/open?id=1sSIYtBY70eY7-WgYawxpYXgOnLHMZeC0ZXfp0kW_0Uo . Sin embargo se verá también en este documento el enfoque o estrategia elegida para el manejo y ejecución de pruebas.

Estrategia

Para asegurar que el software cumple cierto grado de calidad y el manejo de pruebas, se establecieron dos tipos de pruebas: automáticas y manuales. Es importante mencionar que si se encuentra algún error en cualquier tipo de pruebas, se debe notificar inmediatamente al grupo por algún medio virtual. El equipo debe concentrarse a arreglar las anomalías encontradas antes de empezar a implementar alguna otra funcionalidad. Será un proceso guiado por pruebas.

Una gran limitante que se tiene al realizar este tipo de pruebas, es que se necesita un archivo de texto con las absorbancias por longitudes de onda que se generan a partir de un espectrómetro y un software completamente externo al sistema Nitrate. Tales archivos de texto son esenciales para realizar cualquier tipo de prueba ya sea manual o automática.

Pruebas automáticas

Estas se realizarán a diferentes funcionalidades que ejecutará el sistema Nitrate. Se implementarán con Junit 4, ya que se utilizará Java como lenguaje de programación (para la aplicación de escritorio). Estas pruebas automáticas serán de manejo interno, y para la aplicación

de escritorio se realizarán pruebas automáticas solo de: unidad e integración.

Cada vez que se realice una prueba automatizada (ya sea unitaria o de integración), se tendrá que realizar una prueba de regresión en donde se efectúen todas las pruebas que se habían creado anteriormente. Se necesita la mayor cobertura de errores posible, y debido a que son automáticos no repercutirá en el tiempo de desarrollo del sistema Nitrate. Este tipo de pruebas serán de caja blanca, en donde en caso de que no se obtengan los resultados obtenidos, se podrá inspeccionar el código por dentro. También serán estilo *top down*, en donde el nivel y complejidad de pruebas irá aumentando con el tiempo por la metodología iterativa e incremental que se ha escogido.

Pruebas manuales

Serán de tipo de pruebas de usuario. No se necesita ningún tipo de herramienta adicional ni servicio, por lo tanto no es necesario realizar algún tipo de asesoría ni entrenamiento especial antes de efectuar el plan de pruebas. Aplican por igual para tanto la aplicación de escritorio como para celular. Estas pruebas estarán organizadas por caso de uso, en donde se establecerá un caso de prueba para caso de uso.

Para las pruebas de este tipo se utilizará un formato de caja negra, en donde sólo interesan los valores de entrada y los resultados de la prueba. Esta metodología se adapta muy bien al tipo de sistema que se está implementando ya que involucra en mostrar información en pantalla en diferentes formatos: ya sean gráficos o tablas. Se utilizará un enfoque más del lado del usuario. Se realizará las siguientes cantidades de pruebas para cada caso de uso:

- 4 pruebas para los casos de uso con severidad baja.
- 8 pruebas para los casos de uso con severidad media.
- 16 pruebas para los casos de uso de severidad alta.

Es importante mencionar que cada caso de uso será probado de manera aislada y no en grupos. Tampoco se separaran los casos de uso en pequeños grupos, todo será probado en conjunto.

Las pruebas de sistema para la aplicación de escritorio y la aplicación de celular en su totalidad también se realizarán de forma manual. Solo se realizarán pruebas de sistema manuales para la versión de celular ya que como es una aplicación donde su funcionalidad esencial es mostrar gráficos y resultados en pantalla; lo importante serán los resultados desde el punto de

vista exterior de un usuario. Por esta razón solo se toma tal decisión y se adapta a pruebas estilo caja negra.

Con esta estrategia de pruebas, por su gran amplitud y profundidad, se espera encontrar todo tipo de errores. Desde una vista más externa por parte del usuario; como errores del código por dentro que se planean encontrar con las pruebas automáticas de caja blanca. También con este tipo de pruebas se espera localizar errores relacionados con la conexión al servidor y la base de datos. Si los datos mostrados son correctos, significa que se consumió la base de manera correcta. En este caso se tendrá que inspeccionar por dentro los procedimientos almacenados de la base y los *logs* del servidor. Se ha de recalcar que se asume tal suposición de manera indirecta, ya que no hay pruebas específicamente en el uso del servidor y la base de datos.

Resultados del test

Se ha creado un informe de ejecución de pruebas con los resultados del test. En este se especifica: qué pruebas se han realizado, y cuáles no se han realizado (no se planea dejar nada por fuera). Se especifica cuáles módulos se han probado. También hay una lista con los defectos que se han encontrado y se indica si se han arreglado o no. Se incluyen pruebas de tipo en el documento: de unidad automáticas, de integración automáticas, de sistema automáticas (para la aplicación móvil), de sistema manuales (para la aplicación de escritorio), y pruebas manuales para cada caso de uso con sus resultados por cada caso de prueba.

Este informe es sumamente grande como para ponerlo directamente en este documento, por lo cual se ha optado con poner la referencia a este solamente. La versión más reciente de este documento se puede encontrar en:

<https://drive.google.com/open?id=0Bwn9E8E9d8OwcjktTkNZZkxzeTA> .

Reflexión

La sección de reflexión del documento es donde puede hacer generalizaciones partiendo de éxitos o fallos concretos.

Evaluación

Para la realización de este proyecto se utilizó un modelo iterativo incremental, esto quiere decir que las distintas fases del desarrollo del software se realizaron de manera paralela y de forma que el nivel de detalle cada vez fue aumentando más. Haber escogido una metodología de este tipo se considera que fue un éxito, sobre todo por la razón de que no se tenía un concepto muy claro del dominio del problema (área de la química); por lo tanto se recomienda usar esta metodología si no se entiende completamente el software a desarrollar. Desde un punto de vista general se considera que el desarrollo del Sistema Nitrate es un éxito.

La elicitación de requerimientos y las especificaciones de los mismos se considera un éxito. Las reuniones con el cliente y el desarrollo de diagramas, provocaron requerimientos claros y no ambiguos Quedó muy claro el qué el sistema debe hacer. La realización del prototipo también se considera un éxito ya que a partir de él se creó una interfaz de usuario muy amigable. Se contó con una aprobación del prototipo de parte de Laura Hernández.

La fase de diseño también se considera un éxito. Se obtuvo un diseño organizado en capas; con una arquitectura MVC altamente modular considerado el pilar fundamental del sistema Nitrate. Para futuras versiones del proyecto se cree que será un diseño bastante fácil de mantener, extender y reutilizar. El diseño de la base de datos también se considera un éxito, ya que a pesar de que es pequeño y no está normalizado en su mayor nivel; se considera de muy alta eficiencia. Se recomienda que si se prefiere eficiencia por encima a almacenamiento, se tomen decisiones de este tipo. Se obtiene un diseño escalable.

La fase de implementación se considera un éxito también. Toda funcionalidad descrita en el listado de casos de uso fue implementada, y de hecho se implementan funcionalidades extra. Toda revisión de funcionalidad también tuvo una carta de aceptación por parte de Laura Hernández. Se cree que este éxito de diseño fue producto de buenos procesos de diseño y especificación de requerimientos. Se aprovecha muchos conceptos del paradigma orientado a objetos, especialmente en la herencia para formar una jerarquía de clases y permitir comportamientos polimorfismos.

Finalmente la etapa de pruebas también se considera un éxito. Al ser un proceso guiado por pruebas señalar este punto es crítico. Se probó el sistema Nitrate en gran amplitud profundidad, con pruebas de todo tipo: unidad, regresión, integración y del sistema; con formatos de caja negra y caja blanca. Muchas de estas eran automáticas, estableciendo un costo bajo para la realización de pruebas; así como la aplicación de pruebas manuales. Se descubren una cantidad de anomalías producto de la estrategia de pruebas que fue tomada.

Lecciones

Definitivamente se considera haber seleccionado una metodología ágil para el desarrollo del producto como una excelente decisión. Comenzar con toda la elicitación de los requerimientos y luego al diseño, se cree que no se hubiera tenido los mismos resultados. No se conocía con gran detalle lo que se tenía que hacer al principio, y este conocimiento se fue fortaleciendo iteración tras iteración. Si se tiene una segunda oportunidad para realizar el proyecto se volvería a tomar la misma decisión.

Otro factor que se quiere señalar es la implementación de la aplicación móvil estilo *webapp*. Producto de que se tenía poco tiempo, y se quería implementar un sistema altamente portable; se volvería a tomar tal decisión otra vez. Haber construido una aplicación nativa para distintos sistemas operativos hubiera sido muy complicado por el factor tiempo tan limitado que se tenía. Un *webapp* no tendría tal problema, se adapta a cualquier arquitectura, además de que ofrece al usuario actualizaciones instantáneas en todo momento. También haber seleccionado Java para la aplicación de escritorio se considera acertada; por exactamente la misma razón recién mencionada.

Finalmente no hubo ninguna limitación a la hora de realizar el proyecto. A pesar de que no todas las herramientas usadas se conocían, esta no fue una limitante a la hora de realizar el proyecto. El equipo se adaptó rápidamente a estas y a la lógica de negocio.

Errores y limitaciones conocidos

La implementación del sistema Nitrate llegó totalmente a alcanzar la especificación y todo acuerdo realizado con Laura Hernández. Ninguna característica quedó por fuera y todas están presentes. De hecho hay características extra.

Apéndice

Formato

En primer lugar el sistema realizado no funciona por consola, por lo tanto no hay ningún formato de los argumentos de la línea de comando que se debe especificar. Sin embargo, se utiliza un formato JSON para enviar los datos por medio del API y la base de datos. Además, los archivos de proyectos guardados se guardan encriptados bajo la extensión .molabs. El cual sirve de identificador para volver a cargar los proyectos.

También se debe tener en cuenta el formato de los archivos .txt. Estos archivos incluyen toda la información de las absorbancias tomadas de una muestra de agua, ordenadas por longitud de onda. Para que el archivo sea aceptado debe tener tres componentes:

SpectraSuite Data File

+++++

Date: Wed Feb 10 18:52:44 CST 2016

El primero es el título con el nombre de la máquina, después un separador del *header* y al fecha de su creación con el formato observado anteriormente. Posteriormente siguen datos que no son importantes pero al seguir un estándar de la máquina, pueden llegar a utilizarse en el futuro. Estos son:

User: lahernandez

Dark Spectrum Present: Yes

Reference Spectrum Present: Yes

Number of Sampled Component Spectra: 1

Spectrometers: MAYP112250

Integration Time (usec): 100000 (MAYP112250)

Spectra Averaged: 30 (MAYP112250)

Boxcar Smoothing: 0 (MAYP112250)

Correct for Electrical Dark: No (MAYP112250)

Strobe/Lamp Enabled: No (MAYP112250)

Correct for Detector Non-linearity: Yes (MAYP112250)

Correct for Stray Light: No (MAYP112250)

Number of Pixels in Processed Spectrum: 2068

Estos datos no son necesarios, por lo tanto no se necesitan para aceptar el archivo. Después de estos datos deben aparecer las longitudes de onda seguidas de sus absorbancias respectivas, donde las separa un espacio en blanco de la siguiente manera:

189.52 -0.034

189.64 -0.020

189.76 -0.044

189.87 -0.031

189.99 0.101

190.11 0.402

190.23 0.708
190.34 0.774
190.46 0.790
190.58 0.794
190.70 0.799
190.81 0.806

Especificaciones del módulo

El código a implementar siempre seguirá el principio de código limpio “el código no comentado, es el código mejor comentado”. Se seguirán principio de formateo claro para facilitar la lectura del código. Se optará por métodos o funciones cortas con un solo propósito. Se escogen siempre nombres significativos en vez de comentarios. Sin embargo, si hay una sección del código complicada y de difícil lectura, se aceptarán comentarios pero siempre se deberán evitar a su gran mayoría. A la hora de especificar una clase se debe:

Primero se describe la clase en general y se debe realizar un pequeño comentarios acerca de su uso, propósito y funcionamiento general de esta. Luego se escribirán los atributos de esta, de manera de menor a mayor visibilidad. Si es necesario se escribe un comentario especificando su uso en caso de ser necesario. Seguirán los constructores. En caso de que hay a más de uno se deberá comentar su utilidad por separado.

Se sigue por el cuerpo de la clase, compuesto por los métodos. Se deberá ir de la menor visibilidad a la mayor visibilidad; así como de importancia. No es necesario realizar *javadocs* para estos siguiendo algún estándar, pero sí una pequeña descripción de su funcionamiento, entradas y salidas si se considera pertinente.

Para los procedimientos almacenados con extensión .sql; se debe realizar una pequeña documentación acerca del funcionamiento del mismo. También se deben mencionar entradas y salidas.

Casos de prueba

Un banco de pruebas no fue creado para el testing del sistema Nitrate, ni tampoco se realizaron pruebas de estrés. No se realizaron pruebas de entradas enormes. No se probó la escalabilidad de la aplicación. No hubo tiempo para la creación de este tipo de pruebas. Se siguieron las pruebas especificadas en la sección de Pruebas en este documento. Se adjuntan los casos de prueba para las pruebas manuales de cada caso de uso:

Subir archivo de texto (aplicación escritorio)

Id	UC-001
Nombre	Casos de prueba para subir un archivo de texto
Severidad	Alta
Precondiciones	El usuario deberá haber sido autenticado por el sistema Nitrato. El archivo a subir de texto tuvo que haber sido creado externamente.
Poscondiciones	El archivo de texto es cargado al sistema Nitrato y tal situación es mostrada en pantalla.
Pasos y datos	<ol style="list-style-type: none">1. El usuario selecciona la opción de <i>Open File</i> de la pantalla principal.2. El usuario selecciona el archivo que desea subir.
Resultados esperados de cada paso	<ol style="list-style-type: none">1. Se espera que se muestre en pantalla una nueva pantalla con los archivos y directorios del computador.2. Se espera que se muestre en pantalla el archivo cargado y los datos de él en la tabla principal del programa.

Leer archivo de texto (aplicación escritorio)

Id	UC-002
Nombre	Casos de prueba para leer la absorbancia de una archivo de texto
Severidad	Alta
Precondiciones	El usuario deberá haber sido autenticado por el sistema Nitrato. El archivo tuvo que haber sido cargado anteriormente.
Poscondiciones	Se muestra en pantalla la absorbancia de dicho archivo en la tabla principal
Pasos y datos	<ol style="list-style-type: none">1. El usuario ingresa en <i>Wavelength</i> la longitud de onda deseada.2. El usuario selecciona la opción de <i>Absorbance</i>.
Resultados esperados de cada paso	<ol style="list-style-type: none">1. Se muestra en pantalla la longitud de onda ingresada en la fila del archivo.2. Se muestra en pantalla la absorbancia de cada archivo en su respectiva fila.

Generar la fórmula de concentración (aplicación escritorio)

Id	UC-003
Nombre	Casos de prueba para genera la fórmula de concentración
Severidad	Alta
Precondiciones	El usuario deberá haber sido autenticado por el sistema Nitrate. El archivo de texto deberá haber sido cargado anteriormente. Cada archivo deberá poseer una concentración ingresada manual o automáticamente.
Poscondiciones	Se muestra en pantalla la calibración y la generación de la fórmula: intersección con eje y, pendiente y coeficiente <i>Pearson</i> .
Pasos y datos	<ol style="list-style-type: none">1. El usuario selecciona una serie de archivos..2. El usuario selecciona una columna de absorbancia dada.3. El usuario oprime el botón de <i>Calibrate</i>.
Resultados esperados de cada paso	<ol style="list-style-type: none">1. Se muestra en pantalla de manera distintiva los archivos seleccionados.2. Se muestra en pantalla de manera distintiva la columna seleccionada.3. Se muestra en pantalla la nueva calibración en la tabla de calibraciones con su respectiva fórmula.

Observar carpeta (aplicación escritorio)

Id	UC-004
Nombre	Casos de prueba para poner a observar una carpeta
Severidad	Media
Precondiciones	El usuario deberá haber sido autenticado por el sistema Nitrate. La carpeta con sus archivos de texto deberá ser creada anteriormente.
Poscondiciones	La carpeta es observada y todos los archivos en ella cargados
Pasos y datos	<ol style="list-style-type: none">1. El usuario selecciona el submenú de <i>Tools</i> de la pantalla principal.2. El usuario selecciona la opción de <i>Open Observer</i>.3. El usuario selecciona la opción <i>Browse</i> para seleccionar una carpeta.4. El usuario selecciona la carpeta o directorio a observar.5. El usuario selecciona el botón <i>Start</i> para iniciar el observador
Resultados esperados de cada paso	<ol style="list-style-type: none">1. Se espera que se muestra en pantalla las opciones de dicho submenú.2. Se muestra en pantalla la carpeta actualmente seleccionada, o ninguna carpeta en caso de que no se haya seleccionado ninguna anteriormente.3. Se muestra en pantalla una nueva pantalla con los archivos y directorios del computador.4. Se muestra la carpeta seleccionada.5. Se muestra en pantalla los archivos cargados que estaban en la carpeta seleccionada

Calcular el valor de “Sample” (aplicación escritorio)

Id	UC-005
Nombre	Casos de prueba para calcular el valor de Sample
Severidad	Alta
Precondiciones	El usuario deberá haber sido autenticado por el sistema Nitrate. La calibración seleccionada deberá haber sido creada anteriormente con el sistema Nitrate.
Poscondiciones	Se muestra en pantalla el valor de concentración calculado para todas las filas.
Pasos y datos	<ol style="list-style-type: none">1. El usuario selecciona una calibración anteriormente realizada en la tabla de calibraciones.2. El usuario oprime en el botón de <i>Concentration</i>.
Resultados esperados de cada paso	<ol style="list-style-type: none">1. Se muestra en pantalla tal fila de manera distintiva.2. Se muestra en pantalla el valor de la concentración calculado para todas las filas.

Ingresar el valor de concentración estándar (aplicación escritorio)

Id	UC-006
Nombre	Casos de prueba para ingresar el valor de concentración estándar
Severidad	Baja
Precondiciones	El usuario deberá haber sido autenticado por el sistema Nitrate. El archivo de texto deberá haber sido cargado anteriormente
Poscondiciones	Para los archivos seleccionados se debe mostrar en pantalla la concentración manualmente escogida
Pasos y datos	<ol style="list-style-type: none">1. El usuario selecciona en <i>Type</i> el tipo STD para el archivo que desea ingresar la concentración estándar en la tabla principal.2. El usuario ingresa el valor estándar en la columna <i>Concentration</i>.
Resultados esperados de cada paso	<ol style="list-style-type: none">1. Se espera que se muestre en pantalla tal archivo de tipo STD.2. Se muestra en pantalla el archivo seleccionado como STD con su respectiva concentración.

Guardar proyecto (aplicación escritorio)

Id	UC-007
Nombre	Casos de prueba para guardar el estado de un proyecto
Severidad	Alta
Precondiciones	El usuario deberá haber sido autenticado por el sistema Nitrato.
Poscondiciones	El archivo de save es guardado en el computador del usuario.
Pasos y datos	<ol style="list-style-type: none">1. El usuario selecciona el submenú de <i>File</i>.2. El usuario selecciona la opción de <i>Save Project</i>.3. El usuario selecciona un directorio deseado para el archivo <i>save</i>.
Resultados esperados de cada paso	<ol style="list-style-type: none">1. Se muestra en pantalla las opciones de dicho submenú.2. Se muestra en una nueva pantalla con los directorios y archivos del computador del usuario.3. Se crea dicho archivo en el directorio especificado.

Abrir proyecto (aplicación escritorio)

Id	UC-008
Nombre	Casos de prueba para abrir el estado de un proyecto.
Severidad	Alta
Precondiciones	El usuario deberá haber sido autenticado por el sistema Nitrato. El archivo <i>save</i> deberá haber sido creado por la aplicación.
Poscondiciones	El estado completo de un proyecto es cargado y mostrado en pantalla: archivos, concentraciones, absorbancias, calibraciones y configuraciones.
Pasos y datos	<ol style="list-style-type: none">1. El usuario selecciona el submenú de <i>File</i>.2. El usuario selecciona la opción de <i>Open Project</i>.3. El usuario selecciona un archivo <i>save</i> que desea cargar.
Resultados esperados de cada paso	<ol style="list-style-type: none">1. Se muestra en pantalla las opciones de dicho submenú.2. Se muestra en una nueva pantalla con los directorios y archivos del computador del usuario.3. Se muestra en pantalla todo el ambiente de desarrollo. O bien se muestra en pantalla mensaje de error en caso de que el formato del archivo sea incorrecto.

Exportar a Excel (aplicación escritorio)

Id	UC-009
Nombre	Casos de prueba para exportar a Excel
Severidad	Baja
Precondiciones	El usuario deberá haber sido autenticado por el Sistema Nitrate anteriormente. La tabla principal deberá tener al menos una fila.
Poscondiciones	Se crea un archivo Excel en el computador del usuario.
Pasos y datos	<ol style="list-style-type: none">1. El usuario selecciona el submenú de <i>Tools</i>.2. El usuario selecciona la opción de <i>Export Excel</i>.3. El usuario selecciona el directorio en donde desea el archivo Excel.
Resultados esperados de cada paso	<ol style="list-style-type: none">1. Se muestra en pantalla las opciones de dicho submenú.2. Se muestra en una nueva pantalla los directorios y archivos del computador del usuario.3. Se genera el archivo Excel en el directorio especificado.

Generar gráfico concentración - tiempo (aplicación escritorio)

Id	UC-010
Nombre	Casos de prueba para generar el gráfico de concentración vs tiempo
Severidad	Alta
Precondiciones	El usuario deberá haber sido autenticado por el sistema Nitrate. La calibración deberá haber sido creada anteriormente por el sistema Nitrate y debe poseer una concentración.
Poscondiciones	El gráfico de concentración vs tiempo deseado es mostrado en pantalla.
Pasos y datos	<ol style="list-style-type: none">1. El usuario selecciona una serie de filas de la tabla principal.2. El usuario le da click derecho sobre la concentración que desea graficar.3. El usuario selecciona la opción de <i>Graph Concentration</i>.4. El usuario selecciona la pestaña de <i>Concentration</i>.
Resultados esperados de cada paso	<ol style="list-style-type: none">1. Se muestra en pantalla tales filas de manera significativa.2. Se muestra el submenú correspondiente.3. Se muestra en pantalla el gráfico de Concentración vs Tiempo para la calibración seleccionada.4. Se muestra en pantalla el gráfico de Concentración vs Tiempo para la calibración seleccionada.

Generar gráfico absorbancia - concentración (aplicación escritorio)

Id	UC-011
Nombre	Casos de prueba para generar el gráfico de absorbancia vs concentración.
Severidad	Alta
Precondiciones	El usuario deberá haber sido autenticado por el sistema Nitrate anteriormente. La calibración deberá haber sido creada por el Sistema Nitrate anteriormente.
Poscondiciones	El gráfico de absorbancia vs concentración deseado es mostrado en pantalla.
Pasos y datos	<ol style="list-style-type: none">1. El usuario selecciona como <i>active</i> una calibración de la tabla de calibraciones.2. El usuario selecciona la pestaña de <i>Calibration</i>.
Resultados esperados de cada paso	<ol style="list-style-type: none">1. Se muestra en pantalla tal fila de manera distintiva.2. Se muestra en pantalla el gráfico de Absorbancia vs. Concentración.

Exportar gráfico a imagen (aplicación escritorio)

Id	UC-012
Nombre	Casos de prueba para exportar gráfico a imagen
Severidad	Baja
Precondiciones	El usuario deberá haber sido autenticado por el sistema Nitrate.
Poscondiciones	El archivo de imagen es creado en el computador del usuario.
Pasos y datos	<ol style="list-style-type: none">1. El usuario realiza click derecho sobre el gráfico que desea exportar.2. El usuario selecciona la opción de <i>Export image</i>.3. El usuario selecciona el directorio, nombre y extensión para el archivo.4. El usuario selecciona las dimensiones deseadas para la imagen.
Resultados esperados de cada paso	<ol style="list-style-type: none">1. Se muestra las opciones del submenú.2. Se muestra en una nueva pantalla los directorios y archivos del usuario.3. Se muestra una nueva pantalla con las dimensiones deseadas de la imagen.4. El sistema crea dicho archivo de imagen en el directorio especificado.

Generar gráfico concentración vs. tiempo (móvil)

Id	UC-013
Nombre	Casos de prueba para generar el gráfico de concentración vs tiempo
Severidad	Alta
Precondiciones	El usuario deberá haber sido autenticado por el sistema Nitrate. La estación deseada debe existir en el sistema y debe de estar activa.
Poscondiciones	El gráfico de concentración vs tiempo deseado es mostrado en pantalla. Hay una posibilidad que tal gráfico en un momento podría estar vacío.
Pasos y datos	<ol style="list-style-type: none">1. El usuario selecciona una estación dentro del menú de estaciones activas.2. El usuario selecciona la pestaña de <i>Real Time Graphs</i>.3. El usuario hace <i>scroll</i> al gráfico de Concentración vs Tiempo.
Resultados esperados de cada paso	<ol style="list-style-type: none">1. Se muestra en pantalla tal estación de manera distintiva.2. Se muestra en una nueva pantalla los gráficos y datos de dicha estación.3. Se muestra en pantalla dicho gráfico.

Generar gráfico absorbancia vs. concentración (móvil)

Id	UC-014
Nombre	Casos de prueba para generar el gráfico de absorbancia vs concentración
Severidad	Media
Precondiciones	El usuario deberá haber sido autenticado por el sistema Nitrate. La estación deseada debe existir en el sistema y debe de estar activa.
Poscondiciones	El gráfico de absorbancia vs concentración deseado es mostrado en pantalla. También se muestran los datos de la calibración: <i>Pearson</i> , <i>wavelength</i> , longitud de onda e intersección con el eje y
Pasos y datos	<ol style="list-style-type: none">1. El usuario selecciona una estación dentro del menú de estaciones activas.2. El usuario selecciona la pestaña de <i>Real Time Graphs</i>.3. El usuario hace <i>scroll</i> al gráfico de Absorbancia vs Concentración.
Resultados esperados de cada paso	<ol style="list-style-type: none">1. Se muestra en pantalla tal estación de manera distintiva.2. Se muestra en una nueva pantalla los gráficos y datos de dicha estación.3. Se muestra en pantalla dicho gráfico.

Mostrar valor de concentración (móvil)

Id	UC-015
Nombre	Casos de prueba para mostrar el valor de concentración
Severidad	Alta
Precondiciones	El usuario deberá haber sido autenticado por el sistema Nitrate. La estación deseada debe existir en el sistema y debe de estar activa.
Poscondiciones	La última concentración es desplegada en pantalla, de forma significativa.
Pasos y datos	<ol style="list-style-type: none">1. El usuario selecciona una estación dentro del menú de estaciones activas2. El usuario selecciona la pestaña de <i>Concentration Value</i>
Resultados esperados de cada paso	<ol style="list-style-type: none">1. Se muestra en pantalla tal estación de manera distintiva.2. Se muestra en pantalla el valor de concentración de dicha estación.

Exportar gráfico a imagen (móvil)

Id	UC-016
Nombre	Casos de prueba para exportar gráfico a imagen
Severidad	Baja
Precondiciones	El usuario deberá haber sido autenticado por el sistema Nitrate. La estación deseada debe existir en el sistema y debe de estar activa.
Poscondiciones	El gráfico seleccionado es guardado dentro de las fotos del usuario dentro de su dispositivo. Se incluye: gráfico concentración vs tiempo, absorbancia vs concentración y valor de concentración.
Pasos y datos	<ol style="list-style-type: none">1. El usuario hace <i>scroll</i> al gráfico que desea exportar como imagen dentro de la pestaña <i>Real Time Graphs</i>.2. El usuario selecciona el botón de <i>Download</i> junto al gráfico que desea descargar.
Resultados esperados de cada paso	<ol style="list-style-type: none">1. Se muestra en pantalla el gráfico deseado.2. Se guarda el gráfico deseado como archivo de imagen en el dispositivo del usuario.

Glosario

Término	Definición
Absorbancia	Es un concepto físico que corresponde a la medida en que refleja cuánto se atenúa la radiación al pasar por un objeto.
Android	Es un sistema operativo para teléfonos celulares basados en Linux.
Caja blanca	Prueba que además de tomar en cuenta las entradas y salidas; también se centra en la inspección del código fuente del interior. Se prueban distintos flujos de ejecución.
Caja Negra	Prueba en donde solo se toman en cuenta las entradas que recibe y las salidas que produce, sin inspeccionar el funcionamiento interior.
CSS	Es un lenguaje que se utiliza en un ambiente web para crear y asignar los diferentes estilos en una página web.
Dirigido por pruebas	Es un proceso de desarrollo de software en que los requerimientos son convertidos en casos de prueba y las pruebas de unidad son repetidamente ejecutadas en el código fuente. Las pruebas son esenciales.
Eclipse Neon	Es un ambiente de programación para sistemas en Java.
Estación activa	Corresponde a los gráficos y datos de un usuario dado, dentro del cual se ha mantenido activo dentro de los últimos n días en el sistema Nitrade. n podría ser por poner un ejemplo: 3.
HTML	Es un lenguaje estándar para crear la estructura básica en páginas web.
HTTP	Es un protocolo de transferencia de datos basado en hipertexto que se utiliza en el ambiente web.
Incremental	Supone un proceso que el nivel de detalle va del más general al más específico.
iOS	Es un sistema operativo para celular creado por Apple.
Iterativo	Supone un proceso en que se debe realizar muchas repeticiones para lograr el objetivo deseado
Javascript	Es un lenguaje de programación multiparadigma altamente usado en ambientes web y construcción de API's.
Java 8	Versión 8 del lenguaje de programación multiparadigma Java.
JSON	Es un simple archivo, muy entendible para humanos, que sirve como

	intercambio de objetos.
MySQL	Motor de bases de datos relacional.
Nitratos	Es un compuesto químico que es comúnmente encontrado como contaminante en aguas.
nodeJS	Es un intérprete de JavaScript utilizado para la creación de servidores.
Pearson	Es un índice matemático que está entre [-1 y 1] que indica qué tan relacionadas están dos variables dadas.
Pruebas de integración	Son un tipo de pruebas de software que verifica el funcionamiento entre 2 o más componentes de software.
Pruebas de regresión	Son aquel tipo de pruebas que cuando se realiza una prueba, también se tienen que realizar todas las pruebas anteriores a esta.
Pruebas de unidad	Es un tipo de prueba que solamente verifica la funcionalidad de un componente de software o una función.
Polymer	Es una librería escrita en JavaScript código libre creada para construir aplicaciones web utilizando componentes web. Fue desarrollada por Google.
RAM	Es la memoria principal de un computador donde se guardan los programas actualmente en ejecución. Es altamente accesada por el procesador.
Sample	Su traducción directa es muestra. En este caso trata de representar una muestra de agua a la cual se le puede predecir la concentración de nitratos a partir de una calibración.
Servicio Web	Es un conjunto de código de información, que permite la conexión entre varios sistemas o aplicaciones.
STD	Es una abreviación para desviación estándar.
Tiempo Real	Característica que ofrece la visualización de resultados de forma instantánea y en el momento en que ocurren.
Wavelength	Es longitud de onda, y es la distancia entre las ondas de una ola dada.