

Minuta 1.

11/05/2018

Cecilia:

1. Investigue cómo podía validar un correo en Java, encontré que necesitaba de un pattern y un matcher para lograrlo.
2. El problema fue encontrar el pattern correcto ya que existen varios y algunos no me funcionaban.
3. Este problema se puede resolver buscando entre todos los patterns cual es el que contiene todos los caracteres permitidos en un mail

Andrea:

1. Hacer la clase para que obtenga datos de la API y que los formatee de manera que el front pueda utilizarlos
2. Comencé haciendo un script en Python pero el problema era cómo conectarlo con el frontend y con las otras clases
3. Se decidió cambiarlo todo a Java

Miguel:

1. Voy a hacer la clase de java para manejar las alertas y recomendaciones. Va a tener atributos para el símbolo que monitorea y para la acción que recomienda.
2. Tengo un problema definiendo el método para generar las recomendaciones. Necesito saber que parámetros tomar en cuenta y saber si alguno puede ponerse como atributo del objeto.
3. Se decidió utilizar 4 parámetros, el precio de apertura, el promedio de las últimas 24 horas, el precio actual y el cambio actual además, sólo recomienda si el cambio supera una cierta sensibilidad, que va a ser un atributo del objeto.

Monica:

1. Investigación de cómo enviar un correo mediante Mailgun
2. Falta información de la carga de dependencias e importación de clases
3. Investigando más en la documentación de Mailgun para saber las dependencias que faltan

Emiliano:

1. Se inició el proceso de convertir el diseño en una página HTML
2. Todavía hay imágenes que no están definidas para la página
3. Tengo que probar varias imágenes para poder dar una sensación de estilo profesional a la página

Andrés:

1. Investigué cómo crear un servicio rest desde Java

2. Existen varias opciones , pero necesitan la instalación de un servidor para funcionar
3. Hay que investigar como se empaquetaría el proyecto si depende de un servidor third party

Victor

1. Se desarrolló un algoritmo que tuviera la capacidad de poder evaluar las caídas y las subidas del costo de cualquier criptomoneda, de tal manera que pudiera valorar el tamaño y poderle dar un porcentaje de compra o venta
2. Filtrar el lenguaje de acuerdo a una rápida conexión con la librería y un menor uso de código en el despliegue de precios
3. Implementar las pruebas unitarias en todos los métodos fue un reto, ya que algunas variables en el algoritmo se comporta como un diccionario y en otro tipo de ejecución se comportan como floats.

Minuta 2.

16/05/2018

Cecilia:

1. Hice el método para validar el correo, con toda su documentación y pruebas unitarias.
2. Al correr las pruebas unitarias, me daba un error en el log , que no se podían encontrar los appenders del logger y que tenía que inicializar el sistema del log4j correctamente
3. Se solucionó creando un archivo llamado log4j.properties en la carpeta de resources

Andrea:

1. Ya se obtienen los datos de la API por medio de HttpURLConnection y se fijó el intervalo de tiempo a un día.
2. No sé si dejar el número de días fijo o ponerlo como un parámetro.
3. Se decidió agregar un parámetro "lim" al método getData() para que el número de días por los que se obtienen los datos sea flexible.

Miguel:

1. Voy a probar la clase alerta con datos reales, haciendo requests al API de binance.
2. Para eso, necesito saber como hacer requests REST y leer los JSONs de alguna manera.

3. Se encontró una librería llamada OpenUnirest para manejar los requests.

Mónica:

1. Creación de clase java para enviar un correo
2. Errores en el pom por falta de importación de clases
3. Encontrar qué otras clases necesito importar y cuáles son las dependencias extra que necesito

Emiliano:

1. Implementé un scroll que hace ver a la página más dinámica
2. Hay elementos de la página que tienen un comportamiento no deseado con el scroll
3. Hay que reposicionar los elementos para evitar el comportamiento no deseado

Andrés:

1. Decidimos utilizar una clase nativa de java para el servidor debido a que la mayor parte se implementará con web socket.
2. Tengo que saber utilizar las clases httpserver y httphandler
3. Encontre tutoriales y leeré la documentación

Victor

1. Al implementar la prueba unitaria en un método que creaba y eliminar un archivo en el mismo tiempo de ejecución, la prueba limitada al hacer un repaso del código no encontraba el archivo, debido a que fue eliminado y la prueba unitaria no tenía la capacidad de saber que es necesario tener el archivo para poder encontrar en el segundo proceso de su prueba.
2. No se pueden hacer pruebas unitarias en tiempo real en una conexión a una api, debido a que no se sabe que output puede arrojar, debido a la volatilidad de los precios que entrega.
3. Se resolvió programando en todos buscando en cual pudiera cumplir los requisitos filtrados

Minuta 3.

22/05/2018

Cecilia:

1. Subí el método al repo
2. Poder subirlo directamente desde eclipse pero que solo lo subiera a la rama validar y no al master
3. Haciendo un repo de prueba y subirlo ahí hasta encontrar cómo se sube correctamente.

Miguel:

1. Voy a buscar cómo comunicar el backend con el front para poder dar datos sobre precios actuales, así como recomendaciones.
2. Buscar cómo montar un websocket para manejar la comunicación.
3. Hay un ejemplo para montar un websocket para un chat que se va a modificar para nuestro proyecto.

Andrea:

1. Ya se obtienen los datos, se quita lo que no se necesita y se pasa a formato json para que el frontend pueda leerlos.
2. Me di cuenta que el "openTime" está en milisegundos entonces falta cambiarlo a algo que se pueda presentar y que los usuarios entiendan. También falta buscar cómo comunicarse con el frontend.
3. Se encontró una manera de transformar la fecha de milisegundos a formato dd/MM/yyyy y se va a montar todo en un websocket.

Mónica:

1. Corrección en las dependencias y elaboración de pruebas unitarias para enviar correos
2. Existen errores al momento de integrar el modulo de envio de mail con lo demás del proyecto. Los errores solo ocurren en computadoras distintas a la mía.
3. Investigar el error que sucede al momento de correr las pruebas unitarias

Emiliano:

1. Empecé a hacer la gráfica que se va a alimentar del websocket
2. Hay que determinar el formato del json para la comunicación entre front y back
3. Junto con el back se definirá dicho formato

Andrés:

1. Ya pude implementar un primer servidor y una ruta de validación de mail
2. Hay problemas al realizar las rutas del módulo del mail, no logramos compilar todas las partes juntas excepto en la computadora de Mónica
3. Investigar el error al mandar el mail.

Victor:

1. Se puso una condición para que dependiendo como se comportará la variable entrara sin problemas al método.
2. Se sigue investigando como poder determinar la prueba unitaria de tal manera que no borre el archivo hasta que termine todos sus pruebas y al final poderlo borrar cumpliendo con todos los criterios de la prueba unitaria
3. No se puede resolver con la biblioteca de prueba unitaria estándar, debido a eso se tienen una prueba hecha a mano con un historial de precios para así poder determinar el buen funcionamiento del algoritmo.