|  |  |
| --- | --- |
| Clase Prueba (Mario, Adriana) | 1. Incluir el atributo **listAnal,** para guardar los análisis que se definan en las clases. 2. Construir método **ejecutarAnalisis,** que recorra la lista de análisis y ejecute 3. Definición de maquetación del reporte por prueba, que llame a las funciones de **outHtml** de cada clase, de los análisis que tienen. 4. Construcción del método **armarReporte,** que compile el reporte y agregue las diferentes salidas, de acuerdo a lo definido en el punto 4. 5. Definir la función **Prueba,** como inicializador de la clase Prueba, que contenga los parámetros necesarios para la definición de una prueba. |
| Clase Análisis (Mario, Adriana) | 1. Terminar el método **initialize,** quitar la herencia de la clase Prueba. 2. Validar que todos los parámetros del filtro en los análisis estén contemplados en los métodos **loadDictionary** y **loadDatB.** 3. Definir métodos genéricos **outXLSX, outHTML, generarReporte y codeAnalisi**. Con el objetivo de reducir las líneas de código en las clases siguientes. |
| Clase Univariados (William y Robert) | 1. Integración del código **01Omisiones.R** y **02Univariate.R,** para generar un único código que genere las salidas de los dos procesos. 2. Definición de la clase en S4 de acuerdo a los parámetros por defecto del análisis y de acuerdo al modelo de clases. 3. Definir el atributo **outRdata,**  el cual es la ruta del .Rdata, en donde se van a guardar los resultados del análisis. Se deben guardar todas las variables que se usan en las funciones **outHTML y outXLSX.** 4. Separar el código del análisis, del código del reporte en Excel, para construir los métodos **codeAnalisi** y **outXLSX.** 5. Definir las salidas en HTML (si se quiere agregar reportes adicionales de este análisis), función **outHTML.** 6. Probar los métodos implementados para la clase Univariados. |
| Clase TCT (William y Robert) | 1. Lectura y seguimiento del código **03TCT.R**, para conocer el proceso y las salidas de este análisis. 2. Definición de la clase en S4 de acuerdo a los parámetros por defecto del análisis y de acuerdo al modelo de clases. 3. Definir el atributo **outRdata,**  el cual es la ruta del .Rdata, en donde se van a guardar los resultados del análisis. Se deben guardar todas las variables que se usan en las funciones **outHTML y outXLSX.** 4. Separar el código del análisis, del código del reporte en Excel, para construir los métodos **codeAnalisi** y **outXLSX.** 5. Adaptar el código que se había generado para el reporte en HTML del análisis de confiabilidad, construir el método **outHTML** de esta clase**.** 6. Probar los métodos implementados para la clase TCT. |
| Clase Exploratorio(Nelson) | 1. Lectura y seguimiento del código **04Exploratorio.R**, para conocer el proceso y las salidas de este análisis. 2. Definición del método de la clase análisis **calCorrelacion,** que separe el código que se utiliza para hacer el muestreo de los estudiantes y calcular las correlaciones policoricas (Validar función para variables dicotómicas). 3. Definición de la clase en S4 de acuerdo a los parámetros por defecto del análisis y de acuerdo al modelo de clases. 4. Definir el atributo **outRdata,**  el cual es la ruta del .Rdata, en donde se van a guardar los resultados del análisis. Se deben guardar todas las variables que se usan en las funciones **outHTML y outXLSX.** 5. Separar el código del análisis, del código del reporte en Excel, para construir los métodos **codeAnalisi** y **outXLSX.** 6. Definir el reporte HTML de la clase exploratorio, la idea inicial era solamente mostrar el grafico de sedimentación de los diferentes “índices” (bloque, afirmación, escala, etc …) de una prueba y armar un link a las salidas de Excel. Construir el método **outHTML** 7. Probar los métodos implementados para la clase exploratorio. |
| Clase Confirmatorio (Mono) | 1. Lectura y seguimiento del código **05Confirmatorio.R**, para conocer el proceso y las salidas de este análisis. 2. Definición de la clase en S4 de acuerdo a los parámetros por defecto del análisis y de acuerdo al modelo de clases. 3. Definir el atributo **outRdata,**  el cual es la ruta del .Rdata, en donde se van a guardar los resultados del análisis. Se deben guardar todas las variables que se usan en las funciones **outHTML y outXLSX.** 4. Separar el código del análisis, del código del reporte en Excel, para construir los métodos **codeAnalisi** y **outXLSX.** 5. Coordinar con la persona encargada del análisis exploratorio y utilizar en el código la función **calCorrelacion**. 6. Definir el reporte HTML de la clase exploratorio, la idea inicial era solamente mostrar el semPath para loa diferentes modelos de los diferentes “índices” (bloque, afirmación, escala, etc …) de una prueba y armar un link a las salidas de Excel. Construir el método **outHTML** 7. Probar los métodos implementados para la clase confirmatorio. |
| Clase IRT (Mario, Mono, Nelson y Dorita ) | 1. Lectura y seguimiento del código **06IRT.R**, para conocer el proceso y las salidas de este análisis. 2. Definición de la clase en S4 de acuerdo a los parámetros por defecto del análisis y de acuerdo al modelo de clases. 3. Re-organizar el código para tener 3 diferentes métodos (**wrapWINS, wrapPARS, wrapBILOG**) que ejecuten adecuadamente cada uno de los programas utilizados para el análisis de ítem. 4. Separar el código del análisis, del código del reporte en Excel, para construir los métodos **codeAnalisi** y **outXLSX.** 5. Lectura de los resultados de las clases TCT y Univariados, para armar tabla general con toda la información de los ítems y la prueba. 6. Comprobar funciones de ICC, OPC y alertas para los modelos de 1PL y 2PL. 7. Adaptar el código que se había generado para el reporte en HTML del análisis de la aplicación SABER-PRO TyT 2015 y el modelo de 3PL, construir el método **outHTML** de esta clase**.** 8. Probar los métodos implementados para la clase IRT. |
| Clase Invarianza (Dorita) | 1. Lectura y seguimiento del código **07RunInvarianza.R**, para conocer el proceso y las salidas de este análisis. 2. Definición de la clase en S4 de acuerdo a los parámetros por defecto del análisis y de acuerdo al modelo de clases. 3. Definir el atributo **inputFile,** de la clase e incluir este como parámetro en la inicialización de la clase, este atributo debe contener la ruta del archivo en donde se tienen identificado los diferentes grupos poblacionales de la prueba. 4. Definir el atributo **outRdata,**  el cual es la ruta del .Rdata, en donde se van a guardar los resultados del análisis. Se deben guardar todas las variables que se usan en las funciones **outHTML y outXLSX.** 5. Separar el código del análisis, del código del reporte en Excel, para construir los métodos **codeAnalisi** y **outXLSX.** 6. Coordinar con la persona encargada del análisis exploratorio y utilizar en el código la función **calCorrelacion,** esta función deberá ser ejecutada para cada grupo poblacional en donde se quiera tener invarianza. 7. Definir el reporte HTML de la clase invarianza, no se había discutido ninguna maquetación, se debe agregar un link a las salidas de Excel. Construir el método **outHTML** 8. Probar los métodos implementados para la clase de Invarianza. |
| Clase DIFF (Stalyn) | 1. Lectura y seguimiento del código **08RunDIF.R**, para conocer el proceso y las salidas de este análisis. 2. Definición de la clase en S4 de acuerdo a los parámetros por defecto del análisis y de acuerdo al modelo de clases. 3. Definir el atributo **outRdata,**  el cual es la ruta del .Rdata, en donde se van a guardar los resultados del análisis. Se deben guardar todas las variables que se usan en las funciones **outHTML y outXLSX.** 4. Separar el código del análisis, del código del reporte en Excel, para construir los métodos **codeAnalisi** y **outXLSX.** 5. Coordinar con la persona encargada del análisis exploratorio y utilizar en el código la función **calCorrelacion**. 6. Definir el reporte HTML de la clase invarianza, no se había discutido ninguna maquetación, se debe agregar un link a las salidas de Excel. Construir el método **outHTML.** 7. Probar los métodos implementados para la clase DIFF. |