

Herramienta de Formación Virtual

**DOCUMENTO DE DISEÑO DETALLADO**

**PROYECTO DESARROLLO DE UNA HERRAMIENTA DE SOFTWARE BASADA EN TÉCNICAS DE BIG DATA QUE CONTRIBUYA A LA PERMANENCIA DE LOS ESTUDIANTES EN PROCESOS DE EDUCACION VIRTUAL**

**Área de Desarrollo**

|  |
| --- |
| Queda prohibido cualquier tipo de explotación y, en particular, la reproducción, distribución, comunicación pública y/o transformación, total o parcial, por cualquier medio, de este documento sin el previo consentimiento expreso y por escrito de QUEOS S.A.S.  © QUEOS - Derechos Reservados |

Bogotá, DC., Octubre de 2019

**FORMATO PRELIMINAR AL DOCUMENTO**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Título: | **DOCUMENTO DE DISEÑO DETALLADO – VISTAS ARQUITECTÓNICAS** | | | | | | |
| Fecha elaboración | 2019 – 10- 24 | | | | | | |
| Sumario: | Este documento presenta las vistas arquitectónicas de la solución DESARROLLO DE UNA HERRAMIENTA DE SOFTWARE BASADA EN TÉCNICAS DE BIG DATA QUE CONTRIBUYA A LA PERMANENCIA DE LOS ESTUDIANTES EN PROCESOS DE EDUCACION VIRTUAL | | | | | | |
| Palabras Claves: | DISEÑO SOLUCIÓN | | | | | | |
| Formato: | DOC | | Lenguaje: | | | | Español |
| Dependencia: | Área de Desarrollo | | | | | | |
| Código: | QUEOS\_DISENO\_ARQUITECTURA | Versión: | 1.1 | | Estado: | | En desarrollo |
| Categoría: |  | | | | | | |
| Autor (es): | QUEOS | | | Firmas: | |  | |
| Revisó: | Mauricio Chavarro | | |  | |
| Aprobó | Diego Aranda  Gerente de Proyecto  UNAD | | |  | |
| Información Adicional: |  | | | | | | |
| Ubicación: |  | | | | | | |

CONTROL DE CAMBIOS

| **VERSIÓN** | **FECHA** | **No. SOLICITUD** | **RESPONSABLE** | **DESCRIPCIÒN** |
| --- | --- | --- | --- | --- |
| 1.0 | 2019-10-24 |  | QUEOS | Publicación documento de diseño detallado de Vistas arquitectónicas. |
| 1.1 | 2019-11-04 |  | Mauricio H. Chavarro | Revisión y comentarios |
|  | 2020-03-16 |  |  | Actualización de la arquitectura y modelo entidad relación. |

**TABLA DE CONTENIDO**

[1. INTRODUCCIÓN 7](#_Toc22836620)

[1. DISEÑO BASE DE DATOS 8](#_Toc22836621)

[1.1. MODELO ENTIDAD RELACION 9](#_Toc22836622)

[2. ARQUITECTURA DE REFERENCIA 22](#_Toc22836623)

[2.1. VISTA DE PROCESOS 23](#_Toc22836624)

[2.2. VISTA DE IMPLEMENTACION 26](#_Toc22836625)

[2.3. VISTA DE DESPLIEGUE FISICO (DEPLOYMENT) 26](#_Toc22836626)

[2.4. VISTA DE INTEGRACION CON SISTEMAS EXTERNOS 27](#_Toc22836627)

[2.5. VISTA DE PARAMETRIZACION DEL SISTEMA 27](#_Toc22836628)

[3. TERMINOLOGÍA 28](#_Toc22836629)

.

**LISTA DE FIGURAS**

[Figura 14. Diagrama de procesos **¡Error! Marcador no definido.**](#_Toc22826728)

[Figura 15. Diagrama de implementación **¡Error! Marcador no definido.**](#_Toc22826729)

[Figura 16. Diagrama de despliegue **¡Error! Marcador no definido.**](#_Toc22826730)

[Figura 17. Diagrama de integración con sistemas externos **¡Error! Marcador no definido.**](#_Toc22826731)

[Figura 18. Diagrama de parametrización del sistema **¡Error! Marcador no definido.**](#_Toc22826732)

**LISTA DE TABLAS**

No se encuentran elementos de tabla de ilustraciones.

1. INTRODUCCIÓN

E

ste documento presenta el diseño de base de datos y la arquitectura de referencia, preliminar, de acuerdo con el alcance del primer *release*. En este documento se contemplan los siguientes aspectos:

1. Diseño Base Datos: Contiene el diagrama entidad relación del modelo conceptual.
2. Diccionario de Datos: Presenta el diccionario de datos detallado y documentado.
3. Vista de procesos: Describe el diagrama base de cada uno de los procesos de la plataforma.
4. Vista de implementación o componentes: Comprende los componentes y sus relaciones, organizados lógicamente.
5. Vista de despliegue (*deployment*): Describe en detalle la forma en la que los componentes se desplegarán a lo largo de la infraestructura del sistema, por tanto, se definen las capacidades de red requeridas, las especificaciones del servidor, los requisitos de hardware, los requisitos de seguridad, controles de acceso (autenticación y autorización, encriptación, etc.) y otra información relacionada al despliegue del sistema propuesto.
6. Vista de integración con sistemas externos: Especifica el esquema de interoperabilidad de la solución entre sus componentes y con sistemas externos.
7. Vista de parametrización del sistema: Define una lista con todos los parámetros del sistema y una breve descripción de su funcionalidad, se categorizan por subsistemas, procesos, componentes, y visualizaciones e identifica si los parámetros son configurables a nivel de administrador de sistema o de entidad.
8. DISEÑO BASE DE DATOS

El diseño de base de datos presentado en este documento, cubre las historias del primer *release* a desarrollar. Esta sección contiene los siguientes elementos:

* Modelo entidad relación: En este diagrama se grafica cada una de las tablas o entidades y las relaciones entre ellas.
* Diccionario de datos: Detalla cada uno de los atributos, el tipo de dato y una breve descripción del mismo.
* Notas del Diseño: Consideraciones o notas del diseño de base de datos tenidos en cuenta.
  1. MODELO ENTIDAD RELACION

El diagrama entidad-relación, modelo conceptual, despliega las entidades y cómo están relacionados entre sí cada una de estas entidades. Este modelo representa cada una de las entidades con las cuales se representa un conjunto de información y que debe ser almacenada de manera organizada en la base de datos. A continuación se encuentra el modelo entidad relación actualizado a la segunda entrega.



* 1. DICCIONARIO DE DATOS

El diccionario de datos describe las entidades y define de manera detallada el propósito de la información que representa en base de datos cada una de las tablas. En la tabla a continuación se encuentran las tablas para este primer release.

Listado de tablas-entidades del sistema

|  |  |
| --- | --- |
| Nombre Entidad | Descripción |
| CAL\_CALENDARIO | Define los calendarios académicos que va a prestar la universidad para el año. |
| CAR\_CARACTERIZA | Registra la información de caracterización de cada uno de los indicadores calculados, almacena la información calculada del periodo y la información histórica y actual. |
| CARI\_CONSOLIDA\_IND | Almacena todos los indicadores por columnas de manera horizontal por estudiante, año, periodo y los indicadores que definen su comportamiento. Este tipo de almacenamiento agiliza los tiempos de respuesta al momento de consultar la información. |
| CHO\_CAR\_HORAS | Calcula la caracterización por indicador en periodos de tiempo de 2 horas. |
| CUR\_CURSO | Describe los cursos disponibles que tiene la universidad para cada uno de sus programas académicos. |
| DET\_CURSO\_ESTUDIA | Registra por cada estudiante programa académico, calendario académico y curso. |
| EJE\_PROCESO | Tabla de registro de ejecución de procesos por indicador. |
| EST\_ESTUDIANTES | Almacena la información básica de los estudiantes que acceden a educación virtual. |
| IND\_INDICADOR | Registra la definición de los indicadores de caracterización que determinan el comportamiento a calcular por estudiante. |
| MDL\_LOGSTORE\_STANDARD\_LOG | Tabla almacena log transaccional de los estudiantes al interactuar con el CMS. |
| MDL\_MODULO | Registro de los módulos que componen la plataforma. Cada módulo se define con un alias de 3 letras. |
| PAR\_DETG\_DETALLE | Detalle de la definición del grupo de parámetros generales. |
| PAR\_GRU\_GRUPO | Parámetros Grupo - Define encabezado del detalle de los parámetros generales. |
| PER\_PERFIL | Definición y registro de los perfiles o roles. |
| PRG\_PROGRAMA | Registro de cada uno de los programas que conforman la plataforma. |
| PRO\_PROG\_ACADEMICO | Registra los programas académicos virtuales. (Carreras) |
| PXP\_PROG\_PERF | Registro de los programas a los que tiene acceso un perfil o role. |
| UNI\_UNIVERSIDAD | Centro universitario que presta el servicio de educación virtual. |
| USU\_ACCESO | Registra todos los acceso realizados por los usuarios a la plataforma de publicación. |
| USU\_HIST\_CLAVE | Registro de las claves a través de la historia. |
| USU\_USUARIO | Registro de los usuarios que tienen acceso a la plataforma. |

A continuación se van a relacionar cada una de las entidades y el detalle de los atributos que lo componen, especificando su definición y propósito.

|  |  |
| --- | --- |
| Nombre Entidad | CAL\_CALENDARIO |
| Código Entidad | CAL\_CALENDARIO |
| Descripción | Define los calendarios académicos que va a prestar la universidad para ese año. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | **Descripción** |
| Id. Secuencial Calendario | CAL\_ID\_SECUE | numeric(12,0) | Identificador del calendario académico. |
| Id. Secuencial Universidad | UNI\_ID\_SECUE | numeric(12,0) | Registro de la Universidad virtual |
| Nombre Calendario | CAL\_NOMBRE | varchar(200) | Nombre descriptivo del calendario académico definido para el año. |
| Año Calendario | CAL\_AGNO | numeric(5,0) | Año para el cual se definen un grupo de programas académicos. Este es opcional en caso que un calendario cubra mas de un año. |
| Fecha Inicial Calendario | CAL\_FCH\_INICIAL | date | Fecha inicial del calendario para el año. |
| Fecha Final Calendario | CAL\_FCH\_FINAL | date | Fecha final del calendario para el año. |
| Estado Calendario | CAL\_ESTADO | varchar(20) | Estado del calendario académico Activo / Inactivo. Mientras sea vigente el calendario es Activo. De lo contrario es Inactivo. |
| Codigo Calendario | CAL\_CODIGO | varchar(20) | Código del calendario académico con el que se identifica ante la universidad. |
| Descripción | CAL\_DESCRIPCION | varchar(300) | Breve descripción de la definición del calendario. Como parte de la definición se puede relacionar la resolución o acto administrativo. |
| Cod. Calendario Equivalencia | CAL\_CODEQUIVA | varchar(20) | Código de equivalencia entre el log y el registro del sistema central. |
| Semana Inicial Calendario | CAL\_SEMANA\_INI | numeric(11,0) | Semana inicial del calendario académico |
| Semana Final Calendario | CAL\_SEMANA\_FIN | numeric(11,0) | Semana final del calendario académico |

|  |  |
| --- | --- |
| Nombre Entidad | CAR\_CARACTERIZA |
| Código Entidad | CAR\_CARACTERIZA |
| Descripción | Registra la información de caracterización de cada uno de los indicadores calculados, almacena la información calculada del periodo y la información histórica. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | **Descripción** |
| Id. Secuencial Caracterización | CAR\_ID\_SECUE | numeric(12,0) | Identificador secuencial caracterización. |
| Id. Detalle Estudiante | CES\_ID\_SECUE | numeric(12,0) | Registro de los cursos a los cuales ingresa un estudiante. |
| Id. Indicador Secuencial | IND\_ID\_SECUE | numeric(12,0) | Identificador secuencial del indicador generado por la base de datos. |
| Id. Secuencia del proceso | EJE\_ID\_SECUE | numeric(12,0) | Identificador secuencial del proceso. |
| Id. Secuencial Universidad | UNI\_ID\_SECUE | numeric(12,0) | Registro de la Universidad virtual |
| Id. Secuencial Calendario | CAL\_ID\_SECUE | numeric(12,0) | Identificador del calendario académico. |
| Id. Programa Académico | PRO\_ID\_SECUE | numeric(12,0) | Identificador secuencial del programa académico - Carrera Virtual. |
| Id. Curso | CUR\_ID\_SECUE | numeric(12,0) | Identificador secuencial del curso. |
| Id. Estudiante Secuencial | EST\_ID\_SECUE | numeric(12,0) | Identificador Secuencial del estudiante generado por la base de datos |
| Número Período 1 | CAR\_NRO\_PERIODO | numeric(4,0) | Número del periodo en el que se calcula la caracterización. Por ejemplo, número de semana dentro del calendario vigente. |
| Tipo Periodo 1 | CAR\_TIPO\_PERIODO | longtext | Tipo de periodo asociado con el indicador. Diario, Semanal, Mensual, Bimestral, Semestral y/o Anual. |
| Fecha Inicial Periodo | CAR\_FCH\_INICIAL | date | Fecha inicial del periodo de caracterización de donde se toman los logs. |
| Fecha Final Periodo | CAR\_FCH\_FINAL | date | Fecha final del periodo de caracterización de donde se toman los logs. |
| Valor Calculado Periodo 1 | CAR\_VALOR\_PERIODO | decimal(18,6) | Valor calculado y almacenado en esta variable del indicador. Almacena la información exclusiva del periodo. Rango definido por la fecha inicial a la fecha final. |
| Valor Calculado Histórico 1 | CAR\_VALOR\_HISTORIA | decimal(18,6) | Valor calculado y almacenado en esta variable del indicador. Almacena la información del indicador del valor histórico, es decir calculado desde el periodo 1 hasta el periodo n. |
| Año del Cálculo | CAR\_AGNO | numeric(5,0) | Año en que se calcula el indicador de caracterización. El año permite un acceso más rápido a la información. |
| Número logs Procesados 1 | CAR\_NRO\_LOGS | numeric(12,0) | Indica el número de los registros de logs procesados para generar el indicador. |
| Número Periodo Programa | CAR\_NRO\_PER\_PROG | numeric(12,0) | Indica el numero de periodo semana relativo al programa académico. |
| Valor Promedio Hasta Periodo | CAR\_VALOR\_PROMEDIO | decimal(18,6) | Valor promedio calculado a la semana n. |

|  |  |
| --- | --- |
| Nombre Entidad | CHO\_CAR\_HORAS |
| Código Entidad | CHO\_CAR\_HORAS |
| Descripción | Calcula la caracterización por indicador en periodos de tiempo de 2 horas. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | **Descripción** |
| Id. Caracterización Horario | CHO\_ID\_SECUE | numeric(12,0) | Identificador secuencial horario por indicador de una semana o periodo de caracterización. |
| Id. Detalle Estudiante | CES\_ID\_SECUE | numeric(12,0) | Registro de los cursos a los cuales ingresa un estudiante. |
| Id. Secuencia del proceso | EJE\_ID\_SECUE | numeric(12,0) | Identificador secuencial del proceso. |
| Id. Indicador Secuencial | IND\_ID\_SECUE | numeric(12,0) | Identificador secuencial del indicador generado por la base de datos. |
| Id. Secuencial Universidad | UNI\_ID\_SECUE | numeric(12,0) | Registro de la Universidad virtual. |
| Id. Secuencial Calendario | CAL\_ID\_SECUE | numeric(12,0) | Identificador secuencial del calendario académico. |
| Id. Programa Académico | PRO\_ID\_SECUE | numeric(12,0) | Identificador secuencial del programa académico - Carrera Virtual. |
| Id. Curso | CUR\_ID\_SECUE | numeric(12,0) | Identificador secuencial del curso. |
| Número Período | CHO\_NRO\_PERIODO | numeric(4,0) | Número del periodo dentro del calendario del programa académico definido. |
| Fecha Inicial Período | CHO\_FCH\_INICIAL | date | Determina la fecha inicial del periodo evaluado. |
| Fecha Final Período | CHO\_FCH\_FINAL | date | Determina la fecha final del periodo evaluado. |
| Valor Calculado Periodo | CHO\_VLR\_PERIODO | decimal(18,6) | Valor del indicador en el periodo de acuerdo a lo definido en la temporalidad de número de horas y días. |
| Valor Calculado Histórico | CHO\_VLR\_HISTORIA | decimal(18,6) | Valor del indicador histórico de acuerdo a lo definido en la temporalidad de número de horas y días. |
| Hora Inicial | CHO\_HORA\_INICIAL | decimal(18,6) | Indica la hora inicial del rango temporal que está siendo definido. Si es cada 2 horas, será de 0 a 2, de 2 a 4 así sucesivamente hasta 22 a 24. |
| Hora Final | CHO\_HORA\_FINAL | decimal(18,6) | Indica la hora final del rango temporal que está siendo definido. Si es cada 2 horas, será de 0 a 2, de 2 a 4 así sucesivamente hasta 22 a 24. |
| Número Segmento | CHO\_NRO\_SEGMENTO | numeric(4,0) | Indica el número del segmento que representa. Si la longitud del segmento es de 2 horas y hablamos del segmento 2 sería el rango de 2 a 4. |
| Longitud Segmento | CHO\_LON\_SEGMENTO | numeric(4,0) | La longitud del segmento es el número de horas que existe entre la hora final menos la hora inicial. Siguiendo el ejemplo, hablamos 0 a 2, 2 a 4 etc., la longitud es de 2 horas. |
| Día de la semana | CHO\_DIA\_SEMANA | numeric(4,0) | Registra el número del día de la semana en la que se generó el registro log. Siendo Lunes 1 y Domingo 7. |
| Día Hábil | CHO\_TIPO\_DIA | varchar(20) | Registra para este indicador el tipo de día. Entre los tipos de día podemos tener:  1. Día Hábil  2. Día Festivo  3. Sábado  4. Domingo  Con esta definición podemos identificar festivos, sábados y/o domingos. |
| Número Logs Procesados | CHO\_NRO\_LOGS | numeric(12,0) | Indica el número de los registros de log procesados para generar el indicador. |

|  |  |
| --- | --- |
| Nombre Entidad | CUR\_CURSO |
| Código Entidad | CUR\_CURSO |
| Descripción | Registro de los cursos por programa para el calendario académico. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | **Descripción** |
| Id. Curso | CUR\_ID\_SECUE | numeric(12,0) | Identificador secuencial del curso. |
| Id. Secuencial Universidad | UNI\_ID\_SECUE | numeric(12,0) | Registro de la Universidad virtual |
| Nombre del Curso | CUR\_NOMBRE | varchar(150) | Nombre del curso que está siendo definido. Maximo 150 caracteres. |
| Codigo del Curso | CUR\_CODIGO | varchar(20) | Código del curso que está siendo definido. |
| Descripción Curso | CUR\_DESCRI | varchar(500) | Breve descripción del curso. Máximo 500 caracteres. |
| Estado Curso | CUR\_ESTADO | varchar(20) | Estado del curso puede ser Activo-Inactivo |
| Cod. Curso Equivalencia | CUR\_CODEQUIVA | varchar(20) | Código de equivalencia entre el log y el registro del sistema central. |

|  |  |
| --- | --- |
| Nombre Entidad | DET\_CURSO\_ESTUDIA |
| Código Entidad | DET\_CURSO\_ESTUDIA |
| Descripción | Registra por cada estudiante programa académico, calendario académico y curso. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | **Descripción** |
| Id. Detalle Estudiante | CES\_ID\_SECUE | numeric(12,0) | Registro de los cursos a los cuales ingresa un estudiante. |
| Id. Curso | CUR\_ID\_SECUE | numeric(12,0) | Identificador secuencial del curso. |
| Id. Estudiante Secuencial | EST\_ID\_SECUE | numeric(12,0) | Identificador Secuencial del estudiante generado por la base de datos |
| Id. Secuencial Calendario | CAL\_ID\_SECUE | numeric(12,0) | Identificador del calendario académico. |
| Id. Programa Académico | PRO\_ID\_SECUE | numeric(12,0) | Identificador secuencial del programa académico - Carrera Virtual. |

|  |  |
| --- | --- |
| **Nombre Entidad** | EJE\_PROCESO |
| **Código Entidad** | EJE\_PROCESO |
| **Descripción** | Tabla de registro de ejecución de procesos por indicador. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | **Descripción** |
| Id. Secuencia del proceso | EJE\_ID\_SECUE | numeric(12,0) | Identificador secuencial del proceso. |
| Id. Indicador Secuencial | IND\_ID\_SECUE | numeric(12,0) | Identificador secuencial del indicador generado por la base de datos. |
| Año Ejecución | EJE\_AGNO\_EJECUTA | numeric(5,0) | Año de ejecución del proceso para calcular el indicador de caracterización. |
| Fecha Ejecución | EJE\_FCH\_EJECUTA | date | Fecha de ejecución del proceso que calcula el indicador de caracterización. |
| Hora Ejecución | EJE\_HOR\_EJECUTA | time | Hora de ejecución del proceso que calcula el indicador de caracterización. |
| Estado Ejecución | EJE\_ESTADO | varchar(20) | Estado de ejecución que puede ser Pendiente, Proceso, Ejecutado. |
| Nro. Logs Procesado | EJE\_LOGS\_PROCESA | numeric(12,0) | Registra el número de logs procesados en total, para el cálculo. |
| Tiempo Procesamiento | EJE\_TIEMPO\_PROCESO | numeric(5,0) | Tiempo que tarda el proceso en su ejecución. Calculado el tiempo en número de minutos. |
| Nro. Periodo | EJE\_NRO\_PERIODO | numeric(12,0) | Numero del periodo a calcular, correspondiente al año procesado. 52 semanas |
| Tipo Periodo | EJE\_TIPO\_PERIODO | varchar(20) | Tipo de periodo a calcular. De acuerdo a lo establecido debe ser SEMANAL. |
| Año filtro a calcular | EJE\_AGNO | numeric(12,0) | Año de los datos a ejecutar. Filtra el año al cual corresponde los datos de logs |
| Nro. Periodo Calendario | EJE\_NRO\_PER\_PROG | numeric(12,0) | Indica el numero de periodo semana relativo al programa académico. |

|  |  |
| --- | --- |
| **Nombre Entidad** | EST\_ESTUDIANTES |
| **Código Entidad** | EST\_ESTUDIANTES |
| **Descripción** | Almacena la información básica de los estudiantes que acceden a educación virtual. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | **Descripción** |
| Id. Estudiante Secuencial | EST\_ID\_SECUE | numeric(12,0) | Identificador Secuencial del estudiante generado por la base de datos |
| Código Estudiante | EST\_CODIGO | varchar(20) | Código identificador del estudiante generado por MOODLE. |
| Nombre Estudiante | EST\_NOMBRE | varchar(100) | Nombre y apellido del estudiante |
| Cod. Estudiante Equivalencia | EST\_CODEQUIVA | varchar(20) | Código equivalencia del estudiante. Log versus definición sistema central. |

|  |  |
| --- | --- |
| **Nombre Entidad** | IND\_INDICADOR |
| **Código Entidad** | IND\_INDICADOR |
| **Descripción** | Registra los indicadores de caracterización del estudiante. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | **Descripción** |
| Id. Indicador Secuencial | IND\_ID\_SECUE | numeric(12,0) | Identificador secuencial del indicador generado por la base de datos. |
| Nombre Indicador | IND\_NOMBRE | varchar(150) | Nombre del indicador de caracterización que está siendo definido. 150 caracteres. |
| Descripción Indicador | IND\_DESCRIPCION | varchar(500) | Breve descripción del indicador de caracterización definido. 500 Caracteres. |
| Tipo Periodo | IND\_TIPO\_PERIODO | varchar(20) | Define el tipo de periodo para realizar el cálculo, los tipos de periodo pueden ser: Diario, Semanal, Mensual, Bimestral, trimestral, semestral y/o anual. |
| Estado Indicador | IND\_ESTADO | varchar(20) | Define el estado actual del indicador el cual puede ser Activo o Inactivo. Si es Inactivo no se hace el cálculo para el periodo. Si es Activo se genera el cálculo para el periodo. |
| Cobertura Indicador | IND\_COBERTURA | varchar(20) | La cobertura indica que analiza el indicador con base en: Estudiante, Programa, Calendario Académico, Curso y/o Profesor. |
| Codigo Indicador | IND\_INDICADOR | varchar(20) |  |
| Cod. Indicador Equivale | IND\_CODEQUIVA | varchar(20) | Código de equivalencia entre el log y el registro del sistema central. |
| Tipo Indicador | IND\_TIPO\_INDICADOR | varchar(20) | Parámetro tipo de indicador. Ingreso, Tiempo, Otro |

|  |  |
| --- | --- |
| **Nombre Entidad** | MDL\_MODULO |
| **Código Entidad** | MDL\_MODULO |
| **Descripción** | Registro de los módulos que componen la plataforma. Cada módulo se define con un alias de 3 letras. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | **Descripción** |
| Código del Modulo | MDL\_COD | varchar(3) | Código identificador del módulo - Alias de tres letras. |
| Descripción del modulo | MDL\_DESCRI | varchar(100) | Breve descripción del programa que está siendo definido y que forma parte de la plataforma. |

|  |  |
| --- | --- |
| **Nombre Entidad** | PAR\_DETG\_DETALLE |
| **Código Entidad** | PAR\_DETG\_DETALLE |
| **Descripción** | Detalle del grupo de parámetros generales. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | **Descripción** |
| Id. Secuencia detalle | PAR\_DETG\_SECUE | numeric(11,0) | Identificador secuencial del grupo |
| Id. Grupo Parámetro | PAR\_GRU\_SECUE | numeric(11,0) | Identificador secuencial del grupo |
| Codigo detalle | PAR\_DETG\_CODIGO | varchar(20) | Código del grupo de parámetros generales. |
| Código Equivalencia | PAR\_DETG\_CODEQUI | varchar(20) | Nombre del grupo de parámetros |
| Nombre Detalle Parámetro | PAR\_DETG\_NOMBRE | varchar(100) | Nombre del ítem a desplegar que define el ítem |
| Descripción Detalle Parámetro | PAR\_DETG\_DESCRIPCION | varchar(200) | Descripción del ítem detallado de parámetros |
| Estado Parámetro | PAR\_DETG\_ESTADO | varchar(20) | Determina si el ítem es Activo o Inactivo. |

|  |  |
| --- | --- |
| **Nombre Entidad** | PAR\_GRU\_GRUPO |
| **Código Entidad** | PAR\_GRU\_GRUPO |
| **Descripción** | Parámetros Grupo - Define encabezado |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | **Descripción** |
| Id. Grupo Parámetro | PAR\_GRU\_SECUE | numeric(11,0) | Identificador secuencial del grupo |
| Código Grupo Parámetro | PAR\_GRU\_CODIGO | varchar(20) | Código del grupo de parámetros generales. |
| Nombre Grupo parámetro | PAR\_GRU\_NOMBRE | varchar(100) | Nombre del grupo de parámetros |

|  |  |
| --- | --- |
| **Nombre Entidad** | PER\_PERFIL |
| **Código Entidad** | PER\_PERFIL |
| **Descripción** | Definición y registro de los perfiles o roles. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | **Descripción** |
| Código del perfil | PER\_COD | int(11) | Identificador del perfil definido |
| Nombre del perfil | PER\_NOMBRE | varchar(45) | Nombre del perfil o rol definido. |
| Tablas Acceso | PER\_TAB\_ACC | varchar(40) | Condición de la tabla de acceso. |
| SQL Condiciones | PER\_SQL\_ACC | varchar(255) | Condicionamientos asociados al perfil para restringir el acceso a los selects. |

|  |  |
| --- | --- |
| **Nombre Entidad** | PRG\_PROGRAMA |
| **Código Entidad** | PRG\_PROGRAMA |
| **Descripción** | Registro de cada uno de los programas que conforman la plataforma. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | **Descripción** |
| Código del Programa | PRG\_COD | varchar(15) | Código identificador del programa. |
| Descripción del programa | PRG\_DESCRI | varchar(100) | Breve descripción del programa que está siendo definido y que forma parte de la plataforma. |
| Código del Modulo | MDL\_COD | varchar(3) | Código identificador del módulo - Alias de tres letras. |

|  |  |
| --- | --- |
| **Nombre Entidad** | PRO\_PROG\_ACADEMICO |
| **Código Entidad** | PRO\_PROG\_ACADEMICO |
| **Descripción** | Registra los programas académicos virtuales. (Carreras) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | Descripción |
| Id. Programa Académico | PRO\_ID\_SECUE | numeric(12,0) | Identificador secuencial del programa académico - Carrera Virtual. |
| Id. Secuencial Universidad | UNI\_ID\_SECUE | numeric(12,0) | Registro de la Universidad virtual |
| Código Programa Académico | PRO\_COD\_CODIGO | varchar(20) | Código interno con el que se identifica el programa académico. |
| Nombre Programa Académico | PRO\_NOMBRE | varchar(200) | Nombre del programa académico que está siendo definido. |
| Cod. Prog. Equivalencia | PRO\_CODEQUIVA | varchar(20) | Código de equivalencia entre el log y el registro del sistema central. |

|  |  |
| --- | --- |
| **Nombre Entidad** | PXP\_PROG\_PERF |
| **Código Entidad** | PXP\_PROG\_PERF |
| **Descripción** | Registro de los programas a los que tiene acceso un perfil o role. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | **Descripción** |
| Id. Programa Perfil | PXP\_COD | int(11) | Identificador del programa por perfil. |
| Código Programa | PRG\_COD | varchar(15) | Código del programa asociado al perfil. |
| Código Perfil | PER\_COD | int(11) | Código del perfil que asocia un conjunto de programas. |

|  |  |
| --- | --- |
| **Nombre Entidad** | UNI\_UNIVERSIDAD |
| **Código Entidad** | UNI\_UNIVERSIDAD |
| **Descripción** | Centro universitario que presta el servicio de educación virtual. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | **Descripción** |
| Id. Secuencial Universidad | UNI\_ID\_SECUE | numeric(12,0) | Registro de la Universidad virtual |
| Código Universidad | UNI\_CODIGO | varchar(20) | Código o identificador oficial de la universidad virtual. |
| Nombre Universidad | UNI\_NOMBRE | varchar(200) | Nombre de la universidad virtual. |
| Estado Universidad | UNI\_ESTADO | varchar(20) | Estado de la universidad Activo / Inactivo |
| Cod. Equivale Universidad | UNI\_CODEQUIVA | varchar(20) | Código de equivalencia entre el log y el registro del sistema central. |

|  |  |
| --- | --- |
| **Nombre Entidad** | USU\_ACCESO |
| **Código Entidad** | USU\_ACCESO |
| **Descripción** | Registra todos los accesos realizados por los usuarios a la plataforma. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | **Descripción** |
| Identificador del acceso | ID\_SECUE | int(11) | Identificador secuencial de la transacción de acceso. |
| Código del usuario | USU\_COD | varchar(15) | Código identificador del usuario que accede a la plataforma. |
| Fch Acceso | FCH\_ACCESO | date | Fecha de acceso a la plataforma |
| Hora Acceso | HOR\_ACCESO | time | Hora en la que ingreso el usuario a la plataforma |
| Tipo Transacción | TIPO\_TRANSACCION | int(2) | Tipo transacción  1 - Ingreso plataforma  2 - Cambio clave |
| Acceso IP | ACCESO\_IP | varchar(30) | IP de donde accede a la plataforma |

|  |  |
| --- | --- |
| **Nombre Entidad** | USU\_HIST\_CLAVE |
| **Código Entidad** | USU\_HIST\_CLAVE |
| **Descripción** | Registro de las claves a través de la historia. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | **Descripción** |
| Id. Cambio Clave | USU\_HIST\_SECUE | int(11) | Identificador secuencial del cambio de clave |
| Código Usuario | USU\_COD | varchar(15) | Identificador del código de usuario que cambio la clave. |
| Clave Anterior | USU\_CLAVE | varchar(40) | Registra la clave anterior encriptada. |
| Fecha Cambio Clave | USU\_FCH\_CLAVE | date | Fecha en la que el usuario realizo cambio de clave. |

|  |  |
| --- | --- |
| **Nombre Entidad** | USU\_USUARIO |
| **Código Entidad** | USU\_USUARIO |
| **Descripción** | Registro de los usuarios que tienen acceso a la plataforma. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Atributo** | **Código** | **Tipo Dato** | **Descripción** |
| Código del Usuario | USU\_COD | varchar(15) | Código del usuario - Cedula o identificación. |
| Código del Perfil | PER\_COD | int(11) | Identificador del perfil o rol que le da acceso a las opciones habilitadas para dicho perfil. |
| Nombre y Apellido Usuario | USU\_NOMBRE | varchar(60) | Nombre y apellidos del usuario |
| Fch. Creación Usuario | USU\_FCH\_CR | date | Fecha de creación del usuario en el sistema de seguridad y control de acceso. |
| Estado usuario | USU\_ESTADO | varchar(1) | Estado del usuario que determina si está activo o inactivo para ingresar a la plataforma. Activo - Inactivo. |
| Clave usuario | USU\_CLAVE | varchar(40) | Registro de la clave encriptada definida por el usuario. |

A continuación en la siguiente tabla se detalla cada una de las referencias o relaciones existentes entre las entidades que conforman el modelo entidad relación.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LISTADO LLAVES FORANES DEL MODELO ENTIDAD RELACION** | | | | |
| **Nombre Relación** | **Entidad Padre** | **Entidad Hija** | **Columna Llave Foránea** | **Descripción** |
| CALENDARIO\_DETALLE | CAL\_CALENDARIO | DET\_CURSO\_ESTUDIA | Id. Secuencial Calendario | Registra el detalle de ese calendario para el estudiante del programa académico y curso. |
| CARACERIZA\_ESTUDIANTE | DET\_CURSO\_ESTUDIA | CAR\_CARACTERIZA | Id. Detalle Estudiante | Registra la caracterización por estudiante durante un periodo de evaluación y caracterización histórica. |
| CARACTERIZA\_CALENDARIO | CAL\_CALENDARIO | CAR\_CARACTERIZA | Id. Secuencial Calendario | Relaciona el calendario con su caracterización calculado de acuerdo al registro de los log. |
| CARACTERIZA\_CURSO | CUR\_CURSO | CAR\_CARACTERIZA | Id. Curso | Relaciona el curso con su caracterización calculado de acuerdo al registro de los log. |
| CARACTERIZA\_HORAS | DET\_CURSO\_ESTUDIA | CHO\_CAR\_HORAS | Id. Detalle Estudiante | Registra el comportamiento por rango de hora de un indicador para un estudiante durante un periodo de evaluación. |
| CARACTERIZA\_UNIVERSIDAD | UNI\_UNIVERSIDAD | CAR\_CARACTERIZA | Id. Secuencial Universidad | Relaciona la universidad con su caracterización calculado de acuerdo al registro de los log. |
| CURSO\_DETALLE | CUR\_CURSO | DET\_CURSO\_ESTUDIA | Id. Curso | Asocia el curso definido con el estudiante, programa académico y calendario académico |
| EJECUCION\_CALCULAHORAS | EJE\_PROCESO | CHO\_CAR\_HORAS | Id. Secuencia del proceso | Relaciona el proceso de ejecución para calcular los indicadores que caracterizan el comportamiento de los estudiantes en rangos de horas. |
| EJECUCION\_CARACTERIZACION | EJE\_PROCESO | CAR\_CARACTERIZA | Id. Secuencia del proceso | Relaciona el proceso de ejecución para calcular los indicadores que caracterizan el comportamiento de los estudiantes. |
| ESTUDIANTE\_DETALLE | EST\_ESTUDIANTES | DET\_CURSO\_ESTUDIA | Id. Estudiante Secuencial | Relaciona el estudiante con el detalle del curso. |
| HORAS\_CALENDARIO | CAL\_CALENDARIO | CHO\_CAR\_HORAS | Id. Secuencial Calendario | Relaciona el calendario con su caracterización en rango de horas calculado de los log. |
| horas\_curso | CUR\_CURSO | CHO\_CAR\_HORAS | Id. Curso | Relaciona el curso con su caracterización en rango de horas calculado de los log. |
| horas\_programa | PRO\_PROG\_ACADEMICO | CHO\_CAR\_HORAS | Id. Programa Académico | Relaciona programa académico con su caracterización en rango de horas calculado de los log. |
| HORAS\_UNIVERSIDAD | UNI\_UNIVERSIDAD | CHO\_CAR\_HORAS | Id. Secuencial Universidad | Relaciona la universidad con su caracterización en rango de horas calculado de los log. |
| INDICADOR\_CARACTERIZA | IND\_INDICADOR | CAR\_CARACTERIZA | Id. Indicador Secuencial | Registra los indicadores de caracterización del estudiante |
| INDICADOR\_HORARIO | IND\_INDICADOR | CHO\_CAR\_HORAS | Id. Indicador Secuencial | Registra los indicadores calculados de registro horario. |
| perfil\_programa | per\_perfil | pxp\_prog\_perf | Código Perfil | Relaciona el perfil con los programas a los que tiene acceso. |
| proceso\_indicador | IND\_INDICADOR | EJE\_PROCESO | Id. Indicador Secuencial | Relaciona la ejecución del proceso de cálculo con la definición del indicador. |
| programa\_carracterizacion | PRO\_PROG\_ACADEMICO | CAR\_CARACTERIZA | Id. Programa Académico | Relaciona los programas académicos con su caracterización calculado de acuerdo al registro de los log. |
| PROGRAMA\_DETALLE | PRO\_PROG\_ACADEMICO | DET\_CURSO\_ESTUDIA | Id. Programa Académico | Asocia el programa académico con el detalle |
| programa\_modulo | mdl\_modulo | prg\_programa | Código del Modulo | Relaciona con el modulo al cual pertenece dentro de las opciones de la plataforma. |
| programa\_perfil | prg\_programa | pxp\_prog\_perf | Código Programa | Relaciona un programa con los perfiles que tienen acceso. |
| UNIVERSIDAD\_CALENDARIO | UNI\_UNIVERSIDAD | CAL\_CALENDARIO | Id. Secuencial Universidad | Relaciona para la universidad sus diferentes calendarios académicos. |
| UNIVERSIDAD\_CURSO | UNI\_UNIVERSIDAD | CUR\_CURSO | Id. Secuencial Universidad | Asocia por universidad los cursos educativos que presta. |
| UNIVERSIDAD\_PROGRAMAS | UNI\_UNIVERSIDAD | PRO\_PROG\_ACADEMICO | Id. Secuencial Universidad | Relaciona para la universidad los diferentes programas académicos. |
| usuario\_acceso | usu\_usuario | usu\_acceso | Código del usuario | Relaciona el usuario con los accesos que ha tenido a la plataforma. |
| usuario\_clave | usu\_usuario | usu\_hist\_clave | Código Usuario | Relaciona el usuario con los cambios de clave realizados |
| USUARIO\_PERFIL | per\_perfil | usu\_usuario | Código del Perfil | Relaciona el usuario con el perfil que se le asigno. |

* + 1. Notas Generales de diseño base de datos

Este modelo de base de datos tiene un conjunto de consideraciones asociadas con su alcance y la funcionalidad que se requiere, dentro de las consideraciones de diseño tenemos:

1. Tabla temporal log: Se crea una tabla temporal en la base de datos y su propósito es cargar el archivo plano que contiene los logs del periodo para realizar procesos de validación, cargue en tablas de parámetros (estudiantes, cursos, programas entre otros) y cálculo de los indicadores de caracterización para su posterior almacenamiento en las tablas definitivas de almacenamiento de los indicadores. Es importante aclarar que esta tabla no tiene llaves foráneas, con el fin de poder cargar todos los datos del plano en la tabla para su posterior validación, actualización de las tablas de parámetros y finalmente el cálculo. A esta tabla se le definen los índices sobre los atributos que agilicen los cálculos de conteos, sumas, agrupamientos entre otros.
2. Tabla Caracterización: Las tablas de almacenamiento final del cálculo de los indicadores de caracterización, almacenan esta información manteniendo la integridad de la información del detalle de los cálculos con los estudiantes, calendarios, programas académicos y cursos. El nivel de granularidad de ésta permite que los cálculos posteriores sobre los indicadores puedan ser realizados a niveles de agrupación superior.
3. Cargue Tabla Log: Se carga la información directamente a la tabla temporal de log y a partir de allí se debe realizar las siguientes actividades:
   1. Validar obligatoriedad, tipo de datos y rango de valores de cada uno de los atributos.
   2. Verificar si el estudiante, programa y curso está en las tablas de parámetros.
   3. Adicionar los ítems en las tablas de parámetros que no existen y marcarlos como nuevos para su posterior verificación. Así mantenemos la integridad de la información.
   4. Los datos que presenten inconsistencias son registrados en una tabla de inconsistencias, en donde evidencia el ítem y el detalle del error por el cual no fue cargado para procesamiento.
4. ARQUITECTURA DE REFERENCIA

En esta sección se define las diferentes vistas de arquitectónicas de la plataforma, para este primer *release*, presentamos los siguientes componentes:

* Vistas de Procesos: Esta vista argumenta técnicamente cómo la arquitectura va a soportar la operación de la plataforma.
* Vista de Implementación: Muestra gráficamente y describe el empaquetado físico de cada uno de los componentes y sus interacciones.
* Vista de despliegue físico: Describe en detalle la forma en que los componentes se desplegarán a lo largo de la infraestructura del sistema.
* Vista de Integración Componentes Externos: especifica el esquema de interoperabilidad de la solución con otros sistemas de información o entre componentes internos a la plataforma y cómo será dicha interacción en términos de componentes y servicios.
* Vista de parametrización del sistema: Se describe la configuración o parametrización de los diferentes componentes de la plataforma.
  1. VISTA DE PROCESOS

Esta vista argumenta técnicamente cómo la arquitectura va a soportar la operación de la plataforma haciendo uso de los diferentes componentes, su responsabilidad e interconexión. El diagrama de procesos se muestra a continuación:

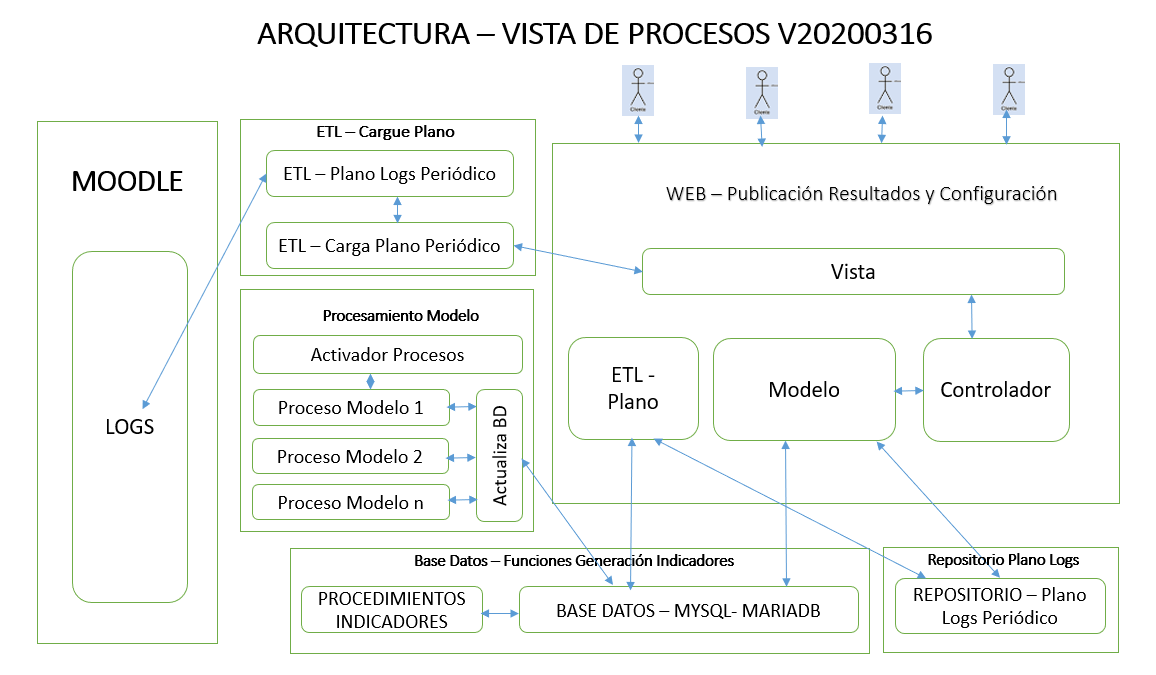


Figura 2. Diagrama Arquitectura de Referencia (MVC)

La plataforma propuesta del diagrama de la figura anterior está compuesto por 5 grandes componentes, que tienen un propósito particular para dar una solución integral. La descripción de cada uno de los componentes y su interacción se describe a continuación:

* **MOODLE**: El componente del recuadro denominado MOODLE genera la información de los logs de todas las actividades e interacciones que realizan los diferentes estudiantes, profesores y administradores sobre dicha plataforma. Esta información queda almacenada en su base de datos, pero para este proyecto se accederá a una muestra que se encuentra en un conjunto de archivos con el detalle de los logs transaccionales para un periodo determinado (El cual debería coincidir con el periodo para el cual se calcularán las métricas de caracterización), dicha muestra refleja el comportamiento transaccional de este grupo de estudiantes. Esta información queda dispuesta en este componente y así será accedida por el componente de extracción y procesamiento.

La idea fundamental es que la HFV no sea invasiva sobre Moodle por razones de confidencialidad de la información del sistema CMS. Lo anterior hace que la interfaz al sistema Moodle sea delgada e independiente de tal forma, que incluso la HFV pueda conectarse en un futuro a plataformas diferentes de Moodle.

* **ETL – Cargue Plano**: Esta es la primera actividad dentro del proceso de modelamiento del comportamiento de los estudiantes, que consiste en la generación del plano de manera periódica y efectuar el cargue en el aplicativo WEB para su posterior procesamiento. Para realizar estas actividades el componente cumple con la siguiente funcionalidad:
  + **ETL – Plano Logs Periódico** : Este componente tiene como función la interacción con la plataforma de aprendizaje virtual para realizar la extracción de los logs periódicos generados por los estudiantes de acuerdo a la estructura estándar. La extracción del plano y la especificación del procedimiento para el alistamiento, generación y cargue del plano de manera periódica es responsabilidad del propietario de la plataforma de aprendizaje virtual (Universidad).

A continuación, en la tabla se encuentra especificada la estructura y el orden de cada uno de los atributos que describen la interacción con la plataforma virtual. Cada uno de los atributos describe el nombre de la columna el tipo de dato y una observación del campo. Es importante que el archivo se genere como se encuentra definido en la tabla, como es un archivo plano los atributos deben ir separados por una barra vertical o pleca “|”.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ESTRUCTURA TABLA DE CARGUE DE LOS LOGS - mdl\_logstore\_standard\_log | | | | |
| id. | Columna Plano | Nombre | Tipo Dato | Observacion |
| 1 | VISAE06| | Programa | String |  |
| 2 | id| | Id | Numérico |  |
| 3 | eventname| | Nombre Evento | String |  |
| 4 | component| | Componente | String |  |
| 5 | ACTION| | Accion | String |  |
| 6 | target| | Objetivo | String |  |
| 7 | objecttable| | Objecttable | String |  |
| 8 | objectid| | Object ID | Numérico |  |
| 9 | crud| | CRUD | String |  |
| 10 | edulevel| | EDULEVEL | Numérico |  |
| 11 | contextid| | CONTEXTID | Numérico |  |
| 12 | contextlevel| | CONTEXTLEVEL | Numérico |  |
| 13 | contextinstanceid| | CONTEXTINSTANCEID | Numérico |  |
| 14 | userid| | USERID | Numérico |  |
| 15 | courseid| | COURSEID | Numérico |  |
| 16 | relateduserid| | RELATEDUSERID | Numérico |  |
| 17 | anonymous| | ANONYMOUS | Numérico |  |
| 18 | timecreated| | TIMECREATED | DateTime | Máximo - 10 caracteres formato LINUX .Se pueden realizar pruebas de la estructura de la fecha hora en el siguiente link - https://www.cdmon.com/es/conversor-timestamp se encontró lo siguiente en los archivos procesados:  1. El formato manejado por Sergio de la info del 2019 para fecha y hora no es correcto ( 20181013160723 ). 2. El formato manejado en el archivo de la información del 2018 es correcto y se puede convertir ejemplo ( 1540908853 ) |
| 19 | origin| | ORIGIN | String |  |
| 20 | ip| | IP | String |  |
| 21 | realuserid | REALUSERID | Numérico |  |

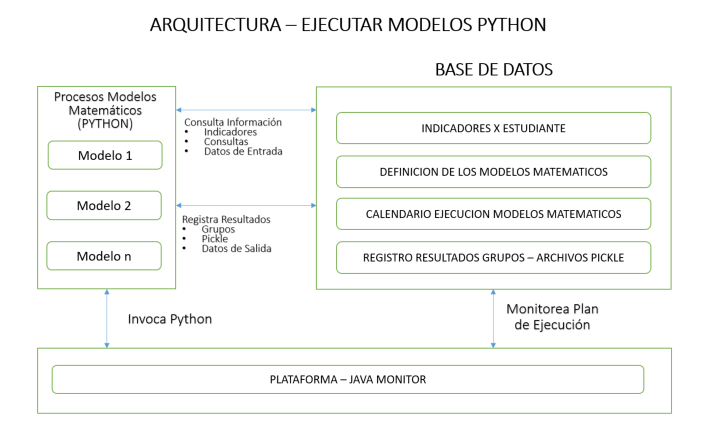
* + **ETL – Carga Plano Periódico** : Una vez generado el plano del año periodo correspondiente, este debe ser cargado a la plataforma WEB de publicación de resultados y Configuración, para lo cual se tiene una opción que selecciona el archivo, realiza este cargue y lo almacena en el repositorio plano de logs, para su posterior validación, procesamiento, respuesta y almacenamiento en base de datos de su detalle transaccional.
  + **Plano Datos Adicionales Parámetros**: El archivo plano se requiere un conjunto de datos adicionales, como son identificador y nombre, esta información se requiere para realizar la actualización de las tablas de parámetros de estudiantes, cursos, profesores y programas académicos, en el proceso de cargue, validación y transformación, las tablas de parámetros y sus relaciones son actualizadas al momento de cargue de los logs y este mismo cargue actualiza las tablas base de parámetros de manera automática evitando tareas adicionales manuales extensas dado el volumen de la data. En la tabla a continuación se encuentra la tabla con los datos adicionales requeridos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DATOS ADICIONALES PLANO | | | | |
| Id | Columna Plano | Nombre | Tipo Dato | Observación |
| 1 | nombreEstudiante |  | String | Nombre y apellido del estudiante |
| 2 | nombreCurso |  | String | Nombre del Curso |
| 3 | IdProgramaAcademico |  | Number | Identificador del programa academico |
| 4 | nombreProgAcademico |  | String | Nombre del programa académico |
| 5 | idProfesor |  | Number | Identificador del profesor |
| 6 | nombreProfesor |  | String | Nombre del profesor |
| 7 | Numero Semana Programa |  | Number | Numero de la semana del programa académico. Este número no corresponde a la semana del año, corresponde al número de semana respecto al inicio del programa académico. |

* **Procesamiento Modelo**: Este componente tiene como objetivo el cálculo de los modelos, para ello cuenta con un conjunto de componentes de software que le permite su activación, así accede a los indicadores, ejecuta los calculo y almacena en base de datos los resultados de los modelos. Una explicación detallada de la funcionalidad de cada uno de los componentes de este módulo es la siguiente:
  + **Activador Procesos**: Este es un componente de software que se ejecuta de manera continua en segundo plano, sin que requiriera de la interacción con los usuarios. Tiene como función activar la ejecución de los Procesos Modelo del 1 al n, en el momento que así lo requieran. En la base de datos se configura el agendamiento de activación de cada uno de estos modelos de acuerdo al momento apropiado para la generación de dichos cálculos. Sus características funcionales son las siguientes:
    - Programa creado en java que se ejecuta de manera continua.
    - Monitorea permanentemente el catálogo de procesos, que se encuentra configurado en la base de datos para activar en la fecha y hora correspondiente el Proceso de Modelo X.
    - Ejecuta el Proceso Modelo X, invocando el programa en Python para que este a su vez se ejecute.
    - Procesamiento Modelo X, Al ser activado el modelo correspondiente este interactúa con la base de datos, procesa la información y el resultado de los modelos son almacenados en base de datos, dejando disponible el resultado de los cálculos para su posterior consulta.
    - Configuración Modelos, cada uno de los modelos especificados deben ser registrados y configurados en la plataforma, así poder realizar su activación programada por la plataforma. Con los modelos en la plataforma se consulta su propósito y características.
  + **Proceso Modelo 1 hasta n**: Son un conjunto de programas que calculan cada uno de los modelos y el resultado es almacenado en base de datos para su publicación y consultas. Cada módulo puede ser creado en lenguajes como PHP, Python y/o java. Cuando es activado este módulo se ejecuta y accede a la base de datos para tomar la información de caracterización de los indicadores por estudiante, con este insumo aplica el modelo y genera una serie de resultados que son almacenados en la base de datos de la plataforma de publicaciones de información.
  + **Actualiza BD**: Con este componente cada uno de los módulos de proceso se debe poder conectar a la base de datos central de la plataforma, para consultar, procesar el modelo y como resultado de dicho proceso la información es registrada en la base de datos.
  + **Consideración Técnica**: Este componente debe ser desarrollado en Python dadas las características de procesamiento matemático, de Ciencias y de Inteligencia Artificial con que cuenta esta herramienta. El procesamiento de la información se realiza en lote de manera periódica de acuerdo con la definición del modelo desarrollado.
* **WEB – Publicación Resultados y Configuración**: **:** Este componente permite la publicación de la información que ha sido extraída, calculada y almacenada en base de datos y la información derivada de los modelos de Data Minning, Business Intellligence e Inteligencia Artificial que se implementen sobre los datos caracterizados. Este componente tiene características técnicas y funcionales que permiten, vía web, acceder a la información de comportamiento del estudiante respecto a su interacción con la plataforma virtual de aprendizaje. Entre sus características y funcionalidades tenemos:
  + **Consideraciones Técnicas**: Esta capa de publicación se desarrolla en Java usando el framework JSF y PrimeFaces, elementos especializados en publicación de información vía WEB. Es un entorno maduro y probado, además de seguir el patrón de desarrollo MVC. Modelo Vista Controlador. Una breve descripción de cada uno de estos componentes es la siguiente:
    - **Vista**: Son el conjunto de interfaces con las que interactúa el usuario para entrada o salida de información.
    - **Modelo**: Contiene la implementación de la lógica del negocio, el acceso y procesamiento de los datos y los mecanismos de almacenamiento en base de datos. Para realizar la conectividad a la base de datos se usa el framework IBATIS que es un framework de código abierto que apoya el proceso de manejo de la persistencia.
    - **Controlador**: Es el intermediario entre la vista y el modelo, dirigiendo y controlando el flujo de información funcional entre el modelo y la vista.
  + **ETL – Plano**: Este es un programa en java que tiene como objetivo abrir, validar y transformar la información de los logs de cada uno de los estudiantes para el periodo y año correspondientes, en caso de ser correctos es cargada esta información de manera estructurada y organizada en una tabla de la base de datos en la que se registra la información para su posterior procesamiento para la generación de los indicadores de dicho periodo o periodos.
* **Base Datos – Funciones Generación Indicadores**: La base de datos es un componente transversal a los componentes de extracción, procesamiento y consulta actual e histórica de los indicadores y modelos, además de mantener el registro de configuración de la plataforma y de las reglas de negocio. Algunas de las características del componente de base de datos y su propósito son los siguientes:
  + **Base Datos – MySQL**: La base de datos propuesta en la arquitectura es una base de datos transaccional MySQL. Que permite el manejo de grandes volúmenes de información con una excelente velocidad de acceso de consulta que facilitan y optimizan el cálculo de los indicadores y de los modelos calculados. Por ser una base de datos relacional mantiene la integridad de la información proveniente de los logs, para el cruce y acceso organizado de la información almacenada, consultada y procesada. La base de datos es un componente transversal a los componentes de extracción y procesamiento con el componente de publicación de resultados y configuración. Algunas de las características del componente de base de datos y su propósito son los siguientes:
    - **Indicadores de Caracterización:** Los resultados son calculados y almacenados para consulta y base para el cálculo de los modelos.
    - **Modelos :** La información de los modelos es almacenada en la base de datos y se deja disponible para consulta.
    - **Publicación Resultados:** La capa de publicación accede en la base de datos a los resultados almacenados y los publica permitiendo acceder a ellos desde diferentes vistas, filtros y diferentes tipos de publicación detallada y/o gráfica.
    - **Configuración de procesamiento:** Cada uno de los indicadores y características de la plataforma son configuradas y almacenadas en la base de datos, para así tener allí registrada esta información requerida para actividades de cálculo de los indicadores, conectividad a los archivos y características propias de la capa de publicación, es decir, parámetros de configuración que determinan la funcionalidad general de la plataforma.
  + **Procedimientos Indicadores**: Es un conjunto de funciones y procedimientos almacenados creados directamente en la base de datos, con los cuales se realizan los conteos, sumatorias, promedios, cálculos requeridos para la generación periódica de cada uno de los indicadores que caracterizan el comportamiento del estudiante al interactuar con la plataforma virtual de aprendizaje. Cada función y/o procedimiento creado en la base de datos calcula un indicador en particular y por estar en la base de datos el tiempo de cálculo es superior al realizar el cálculo en un componente externo a la base de datos. Para su implementación se requiere de conocimientos en SQL y en el lenguaje propietario de la base de datos, el cual es un lenguaje de alto nivel entendible.
* **Repositorio Plano Logs**:
  + **REPOSITORIO – Plano Logs Periódico**: Este componente tiene como función el almacenamiento de los archivos planos con el registro de los logs de la plataforma virtual de aprendizaje, para un año, periodo y estudiante, información detallada base para la generación de los indicadores que describen el comportamiento de este con la plataforma por periodo (Semana unidad de evaluación.), cada uno de estos archivos planos siguen las siguientes actividades:
    - **Cargue del plano**: El archivo plano es ubicado en un directorio especifico de la plataforma, el sistema monitorea permanentemente este directorio, cuando identifique allí la existencia del archivo lo lee, valida, carga a la base de datos y finalmente calcula los indicadores correspondientes de dicho periodo.
    - **Registro del Cargue**: Al momento de realizar el cargue se genera un registro de control en la base de datos dejando rastro de esta actividad de cargue, allí se registra, fecha , hora, usuario que realiza el cargue, año, mes , estado del cargue (Error Estructura, Error Datos, Cargado.), numero de registros procesados, numero de registros cargados, tamaño del archivo plano ,periodo inicial y periodo final. Así podemos consultar la gestión realizada de cada uno de estos archivos planos.
    - **Características del Proceso**: El proceso de cargue y cálculo de los indicadores es un proceso que contempla varias actividades automatizadas. Es importante indicar que se debe cargar por semana los planos. Es decir, en un plano no debe existir información de más de un periodo académico.
* **Implementación Detallada Ejecución Modelos:**

Los modelos matemáticos que actúan sobre la información de los indicadores generados por la plataforma están construidos en Python[[1]](#footnote-2) por parte de los investigadores. Para acceder a la información de entrada a cada modelo efectúan una conexión directa a la base de datos desde el respectivo programa de Python, luego efectúan los procesos de cálculos correspondientes y, finalmente, actualizan en la base de datos de la plataforma los resultados o datos de salida (cálculo de los grupos/categorías por estudiante para el año y periodo), directamente por el proceso del modelo en Python. A continuación se describe la arquitectura y el modelo de datos, para la implementación e interacción de los modelos construidos en Python con la plataforma. También se especifica el paso a paso del procedimiento para el cálculo de los modelos y su integración con la plataforma.

* + **Arquitectura Ejecución Modelos:** A continuación,en la gráfica se ven los componentes, su distribución e interacción, para el proceso de ejecución de los modelos Python. La descripción detallada se encuentra a continuación:



La organización de los componentes para el proceso de ejecución de los modelos matemáticos, se observa en la gráfica de la arquitectura. Allí se pueden ver cada uno de los componentes y su interrelación para llevar a cabo la ejecución de los modelos y el almacenamiento de los resultados en la base de datos. A continuación se explica cada uno de los componentes, su propósito y la relación entre ellos.

**BASE DE DATOS**

Este componente almacena la información que sirve como datos de entrada a los modelos implementados en Python para su cálculo y ejecución entre otros. Para la ejecución de los modelos se dispone de las siguientes características con para su interacción:

* **Indicadores x Estudiante**: La base de datos almacena la información actual e histórica por periodos de los estudiantes calculados por semanas asociadas a un programa académico. Esta información corresponde a los datos de entrada a los procesos de Python para el cálculo de los modelos. Python debe acceder directamente a la base de datos y extraer la información requerida para el cálculo. La información de los indicadores se extrae de la tabla en donde están almacenados los indicadores que corresponde a cari\_consolida\_ind.
* **Definición de los modelos matemáticos**: En la base de datos se define cada uno de los modelos matemáticos definidos por los investigadores y la descripción de cada uno de los grupos que genera como resultado de su ejecución. El investigador una vez con su modelo depurado lo debe registrar en la plataforma así:
  + **Registrar Modelo**: En la plataforma el investigador debe registrar el modelo creado y depurado que va a ser ejecutado en producción de manera periódica, registra su nombre, descripción y el nombre del programa en Python junto con el directorio en donde este va ser almacenado.
  + **Registrar Grupos del Modelo**: Luego ingresa a la plataforma y registra cada uno de los grupos que va a generar su modelo. Así relaciona el modelo matemático y los grupos que va a generar.
* **Calendario Ejecución Modelos Matemáticos**: Una vez con el modelo registrado y los grupos o categorías que lo componen también registrados, debemos realizar la ejecución del modelo matemático, para la ejecución se crea una tarea en un calendario tabla *mexe\_exemodelo*, se indica que el modelo X debe ser ejecutado en una fecha en particular, esta actividad se crea en la plataforma allí, se define el modelo que se va a ejecutar, la fecha de ejecución y para qué año y periodo se deben calcular estos valores de acuerdo a los indicadores previamente definidos. Una vez creado en el calendario esta tarea. La plataforma revisa el momento en que debe ejecutar esta tarea la cual fue programada para una fecha y que se encuentra en estado Pendiente. Cuando se llega a la fecha el sistema activa el programa en Python correspondiente y le envía el identificador del calendario, con este valor se puede acceder al identificador del modelo, a los identificadores de los grupos, el año y periodo de los indicadores base para ejecutar el modelo matemático para ese año y periodo.
* **Registro Resultados Grupos – Archivos Pickle**: Una vez ejecutado el modelo los resultados generados por Python deben ser almacenados en la base de datos directamente por el proceso de Python estableciendo una conexión directa a la base de datos. El aplicativo en Python generado por el investigador accede a la información de la base de datos, así toma la información del modelo, información de los grupos, información de los indicadores para el año y periodo y accede a la información de cada uno de los estudiantes. Una vez en Python calcule por estudiante el grupo/categoría debe almacenar por estudiante en la tabla de resultados *mmes\_modestudiante* el grupo/categoría a la que corresponde dicho estudiante. También en la tabla *mexe\_exemodelo* debe cambiar el estado ha Ejecutado y colocar la información allí requerida asociada a la ejecución. Esta información se encuentra detallada en el diccionario de datos. Esta actualización de información se realiza en el programa en Python, siguiendo los lineamientos de la plantilla que proveemos ModeloV1.py.

**PLATAFORMA JAVA MONITOR**

Este es un componente de la plataforma que monitorea permanentemente el calendario de ejecución de los modelos matemáticos que están implementados en Python por cada uno de los investigadores y cuando llega a la fecha de ejecución y este se encuentra en pendiente, invoca o ejecuta el programa en Python, las actividades realizadas en detalle son las siguientes:

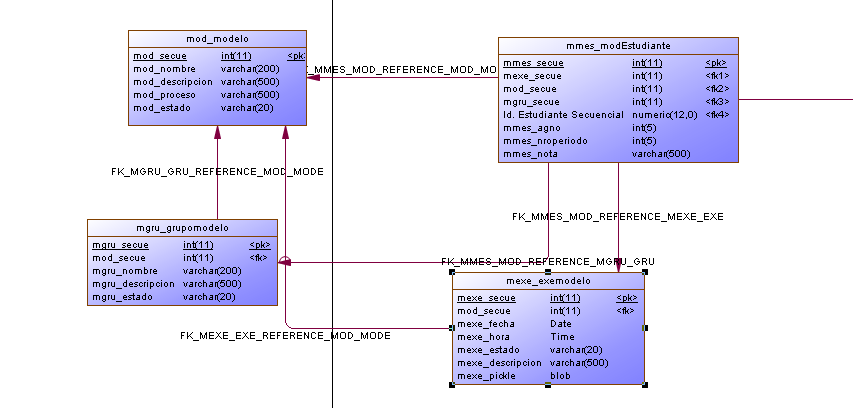
1. Plataforma tiene un programa que monitorea el momento en que debe disparar la ejecución del modelo de acuerdo a la fecha de ejecución programada.
2. Si es la fecha y la tarea del calendario está en pendiente, la plataforma ejecuta el programa en Python (construido por el investigador) que fue definido como parte del modelo.
3. Ejecuta el programa Python y envía como parámetro el identificador del calendario de ejecución del modelo, con este parámetro el programa en Python puede acceder al código del modelo, código de los grupos, código de los estudiantes, al año y numero de periodo. Con esta información es suficiente para buscar en la base de datos la información de los indicadores del año y periodo, para calcular los grupos por estudiante.
4. Ejecuta modelo Python, con esta ejecución debe este programa asociar a cada uno de los estudiantes el grupo y/o categoría y almacenar este resultado en la base de datos de la plataforma.
5. Terminar procesamiento, una vez terminado el procesamiento dentro del programa debe cambiar el estado de la tabla *mexe\_exemodelo* ha Ejecutado y actualizar la información especificada en el diccionario de datos.
6. Los pasos anteriores invocan el modelo, accede a la información de los indicadores, este calcula y almacena en la base de datos, luego cambia el estado del calendario y finaliza la tarea. Integrando así la información de la plataforma con los modelos matemáticos especificados por los investigadores.

**PROCESOS MODELOS MATEMATICOS**

Este es un repositorio en donde se encuentran alojados cada uno de los módulos de Python que implementan los modelos matemáticos. Estos módulos de Python son ejecutados por la plataforma java monitor de acuerdo al calendario de ejecución registrado en la base de datos. Ahora cada módulo de Python en ejecución toma los datos de entrada de la base de datos directamente efectuando una conexión JDBC, luego calculo y los resultados o datos de salida, deben ser almacenados directamente en la base de datos. Como parte de los datos de salida puede ser almacenado en la base de datos si así se requiere el archivo pickle.

* + **Base Datos Modelos**

En la gráfica a continuación se encuentra el modelo entidad relación para el registro de la definición, ejecución, seguimiento , traza y resultados de los modelos matemáticos. El detalle de las tablas y sus atributos se encuentran detallados en la sección diccionario de datos.



* **DICCIONARIO DATOS**

Tabla MEXE\_EXEMODELO

|  |  |
| --- | --- |
| Nombre Tabla | MEXE\_EXEMODELO |
| Descripción | Plan de ejecución de los Modelos. Esta tabla registra cuando y cual modelo ejecutar. |

|  |  |  |
| --- | --- | --- |
| ATRIBUTOS MEXE\_EXEMODELO | | |
| Atributo | Descripción | Tipo de Datos |
| mexe\_secue | Identificador secuencial de la ejecución del modelo matemático | int(11) |
| mod\_secue | Identificador secuencial del modelo que debe ser ejecutado. | int(11) |
| mexe\_fecha | Fecha en que se ejecutó el proceso del modelo matemático. | Date |
| mexe\_hora | Hora en que se ejecutó el proceso del modelo matemático. | Time |
| mexe\_estado | Estado del proceso que ejecuta el Modelo. Pendiente (Para ejecutar), Ejecutado (ya fue Ejecutado), Creado (Tarea Creada) | varchar(20) |
| mexe\_descripcion | Nota de 500 caracteres que la escribe el modelo que ejecuto el proceso. Indicando el estado del resultado. | varchar(500) |
| mexe\_pickle | Este registro que está asociado con la ejecución del modelo matemático almacena, el archivo pickle. | blob |

Tabla MGRU\_GRUPOMODELO

|  |  |
| --- | --- |
| Nombre Tabla | MGRU\_GRUPOMODELO |
| Descripción | Define los grupos calculados de cada uno de los modelos. |

|  |  |  |
| --- | --- | --- |
| ATRIBUTOS MGRU\_GRUPOMODELO | | |
| Atributo | Descripción | Tipo de Datos |
| mgru\_secue | Identificador secuencial de los grupos definidos por cada modelo. | int(11) |
| mod\_secue | Identificador secuencial del modelo al cual pertenece este grupo | int(11) |
| mgru\_nombre | Nombre del grupo definido para el modelo matemático. | varchar(200) |
| mgru\_descripcion | Descripción del grupo calculado por el modelo matemático. Indica que representa este grupo como resultado del modelo. | varchar(500) |
| mgru\_estado | Estado del grupo puede ser Activo / Inactivo | varchar(20) |

Tabla MMES\_MODESTUDIANTE

|  |  |
| --- | --- |
| Nombre Tabla | MMES\_MODESTUDIANTE |
| Descripción | Registra por estudiante, modelo, año y periodo el periodo en que fue clasificado. |

|  |  |  |
| --- | --- | --- |
| ATRIBUTOS MMES\_MODESTUDIANTE | | |
| Atributo | Descripción | Tipo de Datos |
| mmes\_secue | Identificador secuencial de la transacción grupo, modelo y estudiante | int(11) |
| mexe\_secue | Identificador secuencial de la ejecución del modelo matemático | int(11) |
| mod\_secue | Identificador del modelo matemático el cual define el grupo para este estudiante. | int(11) |
| mgru\_secue | Definición del grupo con el que fue clasificado el estudiante por el modelo en esa ejecución. | int(11) |
| Id. Estudiante Secuencial | Identificador Secuencial del estudiante generado por la base de datos | numeric(12,0) |
| mmes\_agno | Año para el que fue calculado y asociado el grupo al estudiante. | int(5) |
| mmes\_nroperiodo | Numero periodo semanal para el que fue calculado y asociado el grupo al estudiante. | int(5) |
| mmes\_nota | Nota especificada por el modelo para este estudiante si requiere aclarar el grupo o resultado. | varchar(500) |

TABLA MOD\_MODELO

|  |  |
| --- | --- |
| Nombre Tabla | MOD\_MODELO |
| Descripción | Definición de los modelos aplicados a los indicadores |

|  |  |  |
| --- | --- | --- |
| ATRIBUTOS MOD\_MODELO | | |
| Atributo | Descripción | Tipo de Datos |
| mod\_secue | Identificador secuencial del modelo definido. | int(11) |
| mod\_nombre | Nombre del modelo matemático. | varchar(200) |
| mod\_descripcion | Descripción del modelo matemático definido y su propósito. | varchar(500) |
| mod\_proceso | Indica el nombre y la ubicación del proceso en Python que ejecuta el modelo matemático para ser invocado por la plataforma. | varchar(500) |
| mod\_estado | Estado del modelo definido puede ser Activo / Inactivo. | varchar(20) |

* + **Tablas a Modificar Modelo Python**

Al ejecutar el modelo matemático en Python, los resultados de dicha ejecución deben seguir el siguiente procedimiento:

* **Modelo Matemático**: El modelo matemático debe ser previamente definido en la tabla de los modelos MOD\_MODELO, En el diccionario de datos se encuentra especificados cada uno de los atributos y su descripción detallada. Si no se matricula este modelo en esta tabla, al registrar los resultados nos genera error de integridad de la información.
* **Grupos del Modelo**: Cada modelo debe tener especificados los grupos resultados de ejecución del modelo, de esta forma, los resultados por cada estudiantes pueden ser registrado en las tabla de resultados. De igual forma que el modelo, los grupos deben ser registrados en la tabla MGRU\_GRUPOMODELO. Previamente a la ejecución cada uno de los grupos deben ser especificados en esta tabla. En caso que no se registren en la base de datos todos los grupos y se adicione un resultado con un grupo que no exista, la base de datos genera un error de integridad de la información.
* **Resultados del Modelo**: El resultado de la clasificación por grupo de los estudiantes, generado como salida del modelo debe ser almacenada directamente por el modelo de Python y esta actualización de la información la debe realizar directamente en la base de datos en la tabla del modelo llamada MMES\_MODESTUDIANTE, la descripción de cada uno de los campos y/o atributos se encuentra especificada en el diccionario de datos.
* **Ejecución del Modelo**: El modelo de Python es ejecutado directamente por un proceso de Java, al momento de realizar el llamado Java le envía un parámetro que corresponde al identificador secuencial de ejecución que se encuentra en la tabla MEXE\_EXEMODELO. Este identificador le permite acceder a la información del registro para realizar las siguientes actividades del proceso:
  + **Identificador del Proceso**: Envía el identificador *mexe\_secue* de la tabla MEXE\_EXEMODELO, este registro le provee a Python el identificador del modelo Python, para que pueda trabajar y actualizar la información de los resultados. También puede acceder a la definición de los grupos, para el registro de los resultados por estudiante.
  + **Registro de los Resultados**: Los resultados de estudiante grupo deben ser almacenados en la tabla MMES\_MODESTUDIANTE. El código de los estudiantes lo debe tomar de la tabla de estudiantes, así asociar el identificador de cada uno de los estudiantes.
  + **Resultados de la Ejecución**: Los resultados de la ejecución, comentarios cambio de estado y el archivo *pickle* asociado, fechas deben ser actualizados en la tabla MEXE\_EXEMODELO.
* **Usuarios**: Para acceder a la plataforma los usuarios acceden a través de la capa de publicación de resultados y configuración. Allí el usuario puede acceder a funcionalidades como:
  + Control de Acceso al aplicativo.
  + Configuración y parametrización de la plataforma.
  + Generación de consultas detalladas.
  + Generación de consultas gráficas.
  + Acceso a la información consolidada por estudiante, curso, programa, profesor entre otros.
  + Consulta de los logs de procesamiento generados.
  1. VISTA DE IMPLEMENTACION

La vista de implementación muestra el empaquetado físico de las partes reutilizables del sistema en unidades sustituibles, llamadas componentes así como sus dependencias.

Los componentes son piezas reutilizables de alto nivel a partir de las cuales se pueden construir los sistemas, además, implementan en la arquitectura física los conceptos y la funcionalidad definidos en la arquitectura lógica. Los componentes pueden ser fuentes, binarios y ejecutables. A continuación se muestra el siguiente diagrama:

[Por desarrollar] Se desarrolla para la segunda entrega.

* 1. VISTA DE DESPLIEGUE FISICO (DEPLOYMENT)

En esta sección se describe en detalle la forma en la que los componentes se desplegarán a lo largo de la infraestructura del sistema. Se detallan las capacidades de red requeridas, las especificaciones del servidor, los requisitos de hardware, los requisitos de seguridad en telecomunicaciones (firewall, bloqueo de puertos, etc.), controles de acceso (autenticación y autorización, encriptación, etc.) y otra información relacionada al despliegue del sistema propuesto.

Por tanto, se define la plataforma tecnológica de hardware, software y comunicaciones que se requiere para implementar y utilizar el sistema propuesto.

[Por desarrollar en 2da. Entrega]

* 1. VISTA DE INTEGRACION CON SISTEMAS EXTERNOS

Este escenario especifica el esquema de interoperabilidad de la solución con otros sistemas de información y cómo será dicha interacción, es decir, quién la genera, en qué eventos se genera y qué datos genera.

[Por desarrollar en 2da. Entrega] Se desarrolla para la segunda entrega.

* 1. VISTA DE PARAMETRIZACION DEL SISTEMA

Esta vista presenta la configuración técnica de los servicios y servidores requeridos para el correcto funcionamiento de la solución en el ambiente de producción.

La siguiente figura presenta la configuración de los servicios de la solución:

[Por desarrollar en 2da. Entrega] Se desarrolla para la segunda entrega.

1. TERMINOLOGÍA

A continuación, se presentan algunos conceptos básicos referentes a los términos más utilizados en el presente documento técnico con el fin de facilitar la comprensión del mismo:

**IBATIS:** Framework para manejo de datos y administración de su persistencia en base de datos.

**REST:** (REST viene de, **RE**presentational **S**tate **T**ransfer), Es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP, para el intercambio de información.

1. Generalmente, llamando librerías desde este lenguaje. [↑](#footnote-ref-2)