**MINUTA**

Minuta No. 005 MGASoluciones

Fecha : Lunes 12 del febrero

Hora : 1:10 am

Lugar : Aula 07, UCR Sede del Atlántico

Siendo las 01:10 am horas del día 12 de febrero de 2016, se reúne el grupo: **InventarioTiendaVirtual**, en sesión extraordinaria previa citación realizada.

**ASISTENTES:**

La persona encargada que precede la reunión, Alberth Calderón Alvarado, pasó lista y determinó que había quórum.

Los asistentes a esta reunión son:

· Albert Calderón Alvarado. \_\_\_\_\_\_Presente\_\_\_\_\_\_\_\_\_

· Anthony Hernández Badilla \_\_\_\_\_\_Presente\_\_\_\_\_\_\_\_\_

· Gustavo Nájera Nájera. \_\_ \_Presente\_\_\_\_\_\_\_\_\_

· Michael Meléndez Mesén. \_\_\_\_\_\_Presente\_\_\_\_\_\_\_\_\_

**AGENDA:**

1. Se asigna tareas Individuales a Alberth Calderon Alvarado.

**ASUNTOS:**

1. **Asuntos tratados y acuerdos tomados**

* Compra de artículos
* Categorías de artículos
* Proveedores
* Un web service para que el proveedor reciba el pedido.

**Tareas concluidas con éxito.**

* 1. **Capa data: Proyecto Inventario**
     1. supplierData:
        1. **insertSupplier($supplier):** Método para insertar un nuevo proveedor en la base de datos, controla que no exista en la base, y le asigna un nuevo número de identificación consecutivo.
        2. **updateSupplier($supplier):** Actualiza los datos de un proveedor en la base de datos.
        3. **desactiveSupplier($supplier):** Desactiva un Proveedor en la base de datos
        4. **activeSupplier():** Activa un Proveedor en la base de datos
        5. **getAllSupplierDesactive():** Obtiene un array de todos los proveedores inactivos
        6. **getAllSupplierActive():**Obtiene un array de todos los proveedores activos
        7. **getAllSupplier():**Obtiene un array de todos los proveedores.
     2. stockData
        1. **insertStock($stock);** registra un artículo en inventario
        2. **stockExist($idProduct,$idStore):** comprueba si el archivo ya existe registrado en la base.
        3. **updateStock($stock):** actualiza un registro del inventario
        4. **getAllStock():** obtiene todos los registros almacenados
        5. **insertExist():** método que obtiene la lista de productos existentes y comprueba si existen registrados en inventario, si el producto no está registrado inserta un nuevo dato de inventario con los valores de cantidad en cero
        6. **getNameProduct($idProduct): obtiene el nombre del producto apartar de su id que posee el dato de inventariogetBrandProduct($idProduct);** obtiene la marca del producto a partir de su id que posee el dato de inventario.
        7. **getModelProduct($idProduct):** obtiene el modelo del producto a partir de su id que posee el dato de inventario
        8. **getNameStore($idProduct**): obtiene el nombre dela tienda a partir de su id que posee el dato de inventario
     3. purchaseData:
        1. **getAllPurchase():** Obtiene todos los registros
        2. **getAllPurchaseSupplier($idSupplier):** obtiene el histórico de compras por proveedor.
  2. **Capa data: webservice**
     1. purchaseData:
        1. **insertPurchase($purchase):** Inserta en la base de datos el registro de un nuevo pedido realizado por la tienda
        2. **getNameProduct($idProduct):** Obtiene el nombre del producto apartar de su id que posee el dato de inventario
        3. **getBrandProduct($idProduct);** Obtiene la marca del producto a partir de su id que posee el dato de inventario.
        4. **getModelProduct($idProduct):** Obtiene el modelo del producto a partir de su id que posee el dato de inventario
        5. **getNameSupplier($idSupplier):** Obtiene el nombre del proveedor atreves de su id.
  3. **Capa Domain Inventario**
     1. **Purchase:** $idPurchases, $idSupplier, $datePurchase, $descriptionPurchase, $idProduct, $totalPurchase, sets y gets
     2. **Stock:** $idStock, $idProduct, $idStore, $idQuantity, $levelStock, sets y gets
     3. **Supplier:** $idSupplier, $nameSupplier, $emailSupplier, $telephoneSupplier, sets y gets
  4. **Capa Domain web service**
     1. **Purchase:** $idPurchases, $idSupplier, $datePurchase, $descriptionPurchase, $idProduct, $totalPurchase, sets y gets
     2. **Stock:** $idStock, $idProduct, $idStore, $idQuantity, $levelStock, sets y gets
     3. **Supplier:** $idSupplier, $nameSupplier, $emailSupplier, $telephoneSupplier, sets y gets
  5. **Capa externa web service**
     1. **PurchaseWeb** Soap
     2. **Scramble** SoapXML
  6. **Capa** **Business**
     1. **SupplierBusiness** controlador de la capa presentación y data de suplier, métodos override
     2. **SupplierActiveAction**, controla la interface de activar proveedores
     3. **supplierinsertAction** controla la inserción de provedores
     4. **supplierupdateAction** controla la actualización de los provedores
     5. **StockBusines** controlador de la capa presentación y data de Stock, métodos override
     6. **stockUpdateAction,**  controla las acciones numéricas actualización y compras.
     7. **Purchase,** controlador de la capa presentación y data purchase, de métodos override
  7. **Capa Presentacion**
     1. **PuchaseHistoryInterface**, Muestra un historial de compras
     2. **StockUpdateInterface** Muestra el estado del inventario de la tienda aca se actualiza y gestionan compras.
     3. **SupplierActiveInterface:** muestra los provedores registrados en condición inactiva y muestra la opción de activarlos.
     4. **SuplierUpdateInterface:** muestra los proveedores activos, se les puede actualizar y desactivar en esta interface.
     5. **SuplierInsertInterface:** Muestra un formulario a completar con los datos que necesita un nuevo proveedor.

• **Tareas a realizar.**

* Automatización de Stock
* Envio idSupplier al webservice

• **Impedimentos**

* Choque de supuestos semánticos para la obtención del proveedor en el web service o en los datos que recibe

**2.** **Tareas realizadas que no fueron planeadas**.

Utilización de Soap, Json, XML, Aprender a utilizarlo, he implementarlos al proyecto.

1. **Asuntos Pendientes**
2. Automatizar Stock conforme a las ventas

**4.** **Asuntos Nuevos**

**5.** **Cumplimiento de tareas asignadas**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tarea** | **Responsable** | **Grado de conclusión** | **Nota** |
| Compra de artículos | Albert Calderón Alvarado | Completo |  |
| CRUD Proveedores | Albert Calderón Alvarado | Completo |  |
| Un web servicie para que el proveedor reciba el pedido. | Albert Calderón Alvarado | Completo |  |

\* \*\*Existen tareas ya definidas como avances revisables en cada iteración:

1. Diagramas de casos de uso

2. Modelo de datos

3. Diagrama de aplicación

4. Módulos funcionales en ejecución

**PRÓXIMA REUNIÓN:**

Fecha: 22 de febrero

Hora: 8:00 AM

Lugar: Lab.C

**CLAUSURA:**

Alberth Calderon Alvarado, como persona encargada de preceder la reunión, clausura la reunión a las 10: 20 horas.

**MINUTA**

Minuta No. 006 MGASoluciones

Fecha : Miecoles 01 de Marzo del 2017

Hora : 3:00 am

Lugar : Laboratorio C, UCR Sede del Atlántico

Se reúne el grupo: **InventarioTiendaVirtual**, en sesión extraordinaria previa citación realizada.

**ASISTENTES:**

La persona encargada que precede la reunión, Alberth Calderón Alvarado, pasó lista y determinó que había quórum.

Los asistentes a esta reunión son:

· Albert Calderón Alvarado. \_\_\_\_\_\_Presente\_\_\_\_\_\_\_\_\_

**AGENDA:**

Se asigna tareas Individuales a Alberth Calderon Alvarado.

**ASUNTOS:**

**Asuntos tratados y acuerdos tomados**

Registro de compras

Registro de compras por Pagar

Registro de Egresos

ReporteMensual de Compras

Reporte Semanal de Compras

Reporte Diario

**Tareas concluidas con éxito.**

* 1. **Capa data: Proyecto Inventario**
     1. supplierData:
        1. **insertSupplier($supplier):** Método para insertar un nuevo proveedor en la base de datos, controla que no exista en la base, y le asigna un nuevo número de identificación consecutivo.
        2. **updateSupplier($supplier):** Actualiza los datos de un proveedor en la base de datos.
        3. **desactiveSupplier($supplier):** Desactiva un Proveedor en la base de datos
        4. **activeSupplier():** Activa un Proveedor en la base de datos
        5. **getAllSupplierDesactive():** Obtiene un array de todos los proveedores inactivos
        6. **getAllSupplierActive():**Obtiene un array de todos los proveedores activos
        7. **getAllSupplier():**Obtiene un array de todos los proveedores.
     2. stockData
        1. **insertStock($stock);** registra un artículo en inventario
        2. **stockExist($idProduct,$idStore):** comprueba si el archivo ya existe registrado en la base.
        3. **updateStock($stock):** actualiza un registro del inventario
        4. **getAllStock():** obtiene todos los registros almacenados
        5. **insertExist():** método que obtiene la lista de productos existentes y comprueba si existen registrados en inventario, si el producto no está registrado inserta un nuevo dato de inventario con los valores de cantidad en cero
        6. **getNameProduct($idProduct): obtiene el nombre del producto apartar de su id que posee el dato de inventariogetBrandProduct($idProduct);** obtiene la marca del producto a partir de su id que posee el dato de inventario.
        7. **getModelProduct($idProduct):** obtiene el modelo del producto a partir de su id que posee el dato de inventario
        8. **getNameStore($idProduct**): obtiene el nombre dela tienda a partir de su id que posee el dato de inventario
     3. purchaseData:
        1. **getAllPurchase():** Obtiene todos los registros
        2. **getAllPurchaseSupplier($idSupplier):** obtiene el histórico de compras por proveedor.
  2. **Capa data: webservice**
     1. purchaseData:
        1. **insertPurchase($purchase):** Actualizacion del metodo que inserta en la base de datos el registro de un nuevo pedido realizado por la tienda, comprueba si debe insertar en compras, o en compras por pagar, ademas de que inserta en la tabla de egresos.
        2. **getNameProduct($idProduct):** Obtiene el nombre del producto apartar de su id que posee el dato de inventario
        3. **getBrandProduct($idProduct);** Obtiene la marca del producto a partir de su id que posee el dato de inventario.
        4. **getModelProduct($idProduct):** Obtiene el modelo del producto a partir de su id que posee el dato de inventario
        5. **getNameSupplier($idSupplier):** Obtiene el nombre del proveedor atreves de su id.
        6. **getIdSupplier($idProduct)**: Obtiene el id del proveedor a partir de su id producto que posee el dato.
  3. **Capa Domain Inventario**
     1. **Purchase:** $idPurchases, $idSupplier, $datePurchase, $descriptionPurchase, $idProduct, $totalPurchase, sets y gets
     2. **Stock:** $idStock, $idProduct, $idStore, $idQuantity, $levelStock, sets y gets
     3. **Supplier:** $idSupplier, $nameSupplier, $emailSupplier, $telephoneSupplier, sets y gets
  4. **Capa Domain web service**
     1. **Purchase:** $idPurchases, $idSupplier, $datePurchase, $descriptionPurchase, $idProduct, $totalPurchase, sets y gets
     2. **Stock:** $idStock, $idProduct, $idStore, $idQuantity, $levelStock, sets y gets
     3. **Supplier:** $idSupplier, $nameSupplier, $emailSupplier, $telephoneSupplier, sets y gets
  5. **Capa externa web service**
     1. **PurchaseWeb** Soap
     2. **Scramble** SoapXML
  6. **Capa** **Business**
     1. **SupplierBusiness** controlador de la capa presentación y data de suplier, métodos override
     2. **SupplierActiveAction**, controla la interface de activar proveedores
     3. **supplierinsertAction** controla la inserción de provedores
     4. **supplierupdateAction** controla la actualización de los provedores
     5. **StockBusines** controlador de la capa presentación y data de Stock, métodos override
     6. **stockUpdateAction,**  controla las acciones numéricas actualización y compras.
     7. **Purchase,** controlador de la capa presentación y data purchase, de métodos override
  7. **Capa Presentacion**
     1. **PuchaseHistoryInterface**, Muestra un historial de compras
     2. **StockUpdateInterface** Muestra el estado del inventario de la tienda aca se actualiza y gestionan compras.
     3. **SupplierActiveInterface:** muestra los provedores registrados en condición inactiva y muestra la opción de activarlos.
     4. **SuplierUpdateInterface:** muestra los proveedores activos, se les puede actualizar y desactivar en esta interface.
     5. **SuplierInsertInterface:** Muestra un formulario a completar con los datos que necesita un nuevo proveedor.

• **Tareas a realizar.**

* Automatización de Stock
* Envio idSupplier al webservice

• **Impedimentos**

* Choque de supuestos semánticos para la obtención del proveedor en el web service o en los datos que recibe

**2.** **Tareas realizadas que no fueron planeadas**.

Utilización de Soap, Json, XML, Aprender a utilizarlo, he implementarlos al proyecto.

1. **Asuntos Pendientes**
2. Automatizar Stock conforme a las ventas

**4.** **Asuntos Nuevos**

**5.** **Cumplimiento de tareas asignadas**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tarea** | **Responsable** | **Grado de conclusión** | **Nota** |
| Compra de artículos | Albert Calderón Alvarado | Completo |  |
| CRUD Proveedores | Albert Calderón Alvarado | Completo |  |
| Un web servicie para que el proveedor reciba el pedido. | Albert Calderón Alvarado | Completo |  |

\* \*\*Existen tareas ya definidas como avances revisables en cada iteración:

1. Diagramas de casos de uso

2. Modelo de datos

3. Diagrama de aplicación

4. Módulos funcionales en ejecución

**PRÓXIMA REUNIÓN:**

Fecha: 22 de febrero

Hora: 8:00 AM

Lugar: Lab.C

**CLAUSURA:**

Alberth Calderon Alvarado, como persona encargada de preceder la reunión, clausura la reunión a las 10: 20 horas.