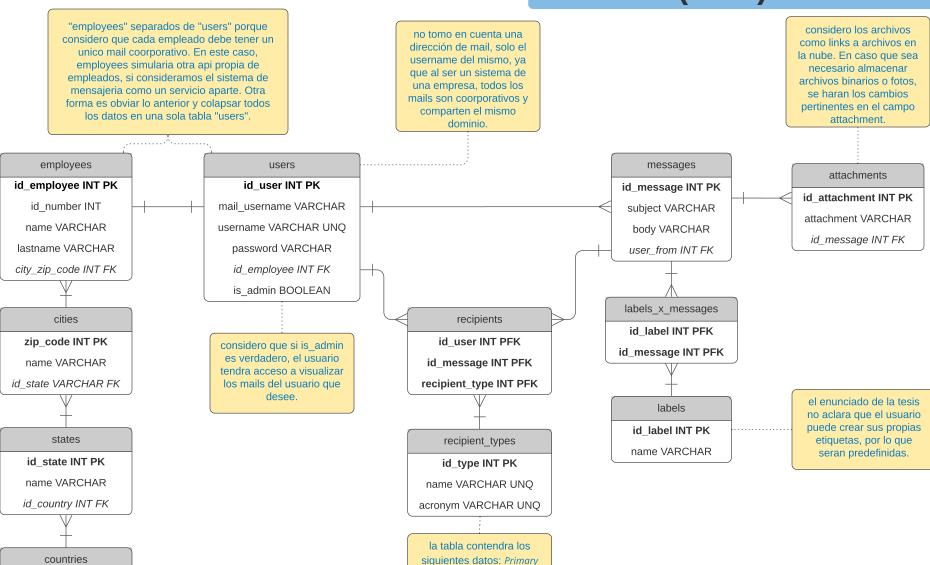
ERD (UML) - Final Thesis Work - V1



Receptor(To) ,Carbon

Copy(CC) orBlind Carbon Copy(BCC)

id_country INT PK

name VARCHAR UNQ

Final Thesis work

This is your final thesis to be a certified Glober in Java Web Services.

We need to create a Web Service to allow a company handle a internal messages between employees. Our web service should give support to the following functional requirements:

- 1. User registration: We should have the following data: username, name, last name, identification number, address, zip code city, state, country
- 2. User login: Web service have to have the ability to login a registerd user to send and receive messages.
- Message sending: Web service should allow to send messages to multiple registered users specifying each recipient as a Primary Receptor (To), Carbon Copy (CC) or Blind Carbon Copy (BCC). User should be logged to sendand receive messages messages. Messages should have the following data: Subject, Body and Attachments.
- 4. Message Reception: Web service should allow to receive messages. The user can receive message and catalogue each one with one or more labels.
- 5. Inbox and Sent: Web service will provide support to query messages in inbox and sent.
- 6. Label Filtering: Web service must provide tools to filter messages by label. User can add different labels to each message. Each user can create their own personalized Web Services.

Our Web Service should meet the following Non-Functional requirements:

- RESTFUL: Web service should comply the REST principles.
- 2. Pagination: Web service must allow pagination for queries.
- 3. Performance: Web service must respond each in less than 10 milliseconds
- 4. Use design patterns and good practices: You must use Design patterns learned in this course.
- 5. Testing: You must reach at 80% of Unit Testing coverage in your solution.

You can use the storage you consider the best to provide support to solve this problem. We prefer to use a Relational Database , but if you consider if a NoSQL database fits to comply requirements it's ok.

Optional Activity

In order to have a more comprehensive picture on how Web Services works in a *normal* environment, you can Dockerize the solution creating containers in an internal network for DB and Web Service and connect both.

ERD (UML) - Final Thesis Work - V2 "employees" separados de "users" porque (Correción) la considero que cada empleado debe tener un columna unico mail coorporativo. En este caso, considero los archivos no tomo en cuenta una mail username pasa employees simularia otra api propia de como links a archivos en dirección de mail, solo el a la tabla empleados, si consideramos el sistema de la nube. En caso que sea username del mismo, ya Employees, porque mensajeria como un servicio aparte. Otra necesario almacenar que al ser un sistema de (tomando como forma es obviar lo anterior y colapsar todos archivos binarios o fotos. una empresa, todos los eiemplo Globant) el los datos en una sola tabla "users". se haran los cambios mails son coorporativos v mail es predefinido pertinentes en el campo comparten el mismo para el empleado. attachment dominio. employees users messages attachments id_employee INT PK id user INT PK id_message INT PK id attachment INT PK username VARCHAR UNO id number INT subject VARCHAR attachment VARCHAR name VARCHAR password VARCHAR body VARCHAR id message INT FK lastname VARCHAR id_employee INT FK user_from INT FK city zip code INT FK is admin BOOLEAN mail username VARCHAR labels x messages labels_x_user recipients cities id label INT PFK id label INT PFK id user INT PFK considero que si is admin id_message INT PFK zip code INT PK id user INT PFK es verdadero, el usuario id message INT PFK tendra acceso a visualizar name VARCHAR recipient_type INT PFK los mails del usuario que id state VARCHAR FK desee. states recipient types labels id state INT PK id_type INT PK id label INT PK name VARCHAR name VARCHAR UNO name VARCHAR id country INT FK acronvm VARCHAR UNO la tabla contendra los countries (Correción) Cada usuario siguientes datos: Primary puede crear sus propios id country INT PK Receptor(To), Carbon labels. Copy(CC) orBlind Carbon name VARCHAR UNQ Copy(BCC)

Final Thesis work

This is your final thesis to be a certified Glober in Java Web Services.

We need to create a Web Service to allow a company handle a internal messages between employees. Our web service should give support to the following functional requirements :

- 1. User registration: We should have the following data: username, name, last name, identification number, address, zip code city, state, country
- 2. User login: Web service have to have the ability to login a registerd user to send and receive messages.
- 3. Message sending: Web service should allow to send messages to multiple registered users specifiying each recipient as a Primary Receptor (To), Carbon Copy (CC) or Blind Carbon Copy (BCC). User should be logged to sendand receive messages messages should have the following data: Subject, Body and Attachments,
- 4. Message Reception: Web service should allow to receive messages. The user can receive message and catalogue each one with one or more labels.
- 5. Inbox and Sent: Web service will provide support to query messages in inbox and sent.
- 6. Label Filtering: Web service must provide tools to filter messages by label. User can add different labels to each message. Each user can create their own personalized Web Services.

Our Web Service should meet the following Non-Functional requirements:

- 1. RESTFUL: Web service should comply the REST principles.
- 2. Pagination: Web service must allow pagination for queries.
- Performance: Web service must respond each in less than 10 milliseconds
- 4. Use design patterns and good practices: You must use Design patterns learned in this course.
- 5. Testing: You must reach at 80% of Unit Testing coverage in your solution.

You can use the storage you consider the best to provide support to solve this problem. We prefer to use a Relational Database , but if you consider if a NoSQL database fits to comply requirements it's ok.

Optional Activity

In order to have a more comprehensive picture on how Web Services works in a normal environment, you can Dockerize the solution creating containers in an internal network for DB and Web Service and connect both.

Comentario de Nacho a la V1:

• Labels: El usuario puede crear sus propios labels. Solo que hay un typo en el enunciado del ejercicio (Igual si son predefinidos, no va a estar mal tampoco)

ERD (UML) - Final Thesis Work - V1 "employees" separados de "users" porque (Correción V2) la considero que cada empleado debe tener un columna unico mail coorporativo. En este caso, considero los archivos no tomo en cuenta una mail username pasa employees simularia otra api propia de como links a archivos en dirección de mail, solo el a la tabla empleados, si consideramos el sistema de la nube. En caso que sea username del mismo, ya Employees, porque mensajeria como un servicio aparte. Otra necesario almacenar que al ser un sistema de (tomando como forma es obviar lo anterior y colapsar todos archivos binarios o fotos. una empresa, todos los eiemplo Globant) el los datos en una sola tabla "users". se haran los cambios mails son coorporativos v mail es predefinido pertinentes en el campo comparten el mismo para el empleado. attachment dominio. employees users messages attachments id_employee INT PK id user INT PK id_message INT PK id attachment INT PK username VARCHAR UNO id number INT subject VARCHAR attachment VARCHAR name VARCHAR password VARCHAR body VARCHAR id message INT FK lastname VARCHAR id_employee INT FK user from INT FK city zip code INT FK is admin BOOLEAN mail username VARCHAR recipients labels labels x messages cities id user INT PFK did_label_x_message INT PK id label INT PK considero que si is admin zip code INT PK es verdadero, el usuario id label INT PFK id message INT PFK name VARCHAR tendra acceso a visualizar name VARCHAR id message INT PFK recipient_type INT PFK los mails del usuario que id user INT PFK id state VARCHAR FK desee. (Correción V2) states recipient types (Correción V3) No Cada usuario puede vendran los label por id state INT PK id_type INT PK crear sus propios defecto, los debe labels. name VARCHAR name VARCHAR UNO crear el usuario. Ademas, la tabla id country INT FK acronvm VARCHAR UNO tendra su propio idenificador para pode crear su la tabla contendra los repositorio. countries siguientes datos: Primary

Receptor(To), Carbon Copy(CC) orBlind Carbon

Copy(BCC)

id country INT PK

name VARCHAR UNQ

Final Thesis work

This is your final thesis to be a certified Glober in Java Web Services.

We need to create a Web Service to allow a company handle a internal messages between employees. Our web service should give support to the following functional requirements:

- 1. User registration: We should have the following data: username, name, last name, identification number, address, zip code city, state, country
- 2. User login: Web service have to have the ability to login a registerd user to send and receive messages.
- 3. Message sending: Web service should allow to send messages to multiple registered users specifiying each recipient as a Primary Receptor (To), Carbon Copy (CC) or Blind Carbon Copy (BCC). User should be logged to sendand receive messages messages should have the following data: Subject, Body and Attachments,
- 4. Message Reception: Web service should allow to receive messages. The user can receive message and catalogue each one with one or more labels.
- 5. Inbox and Sent: Web service will provide support to query messages in inbox and sent.
- 6. Label Filtering: Web service must provide tools to filter messages by label. User can add different labels to each message. Each user can create their own personalized Web Services.

Our Web Service should meet the following Non-Functional requirements:

- 1. RESTFUL: Web service should comply the REST principles.
- 2. Pagination: Web service must allow pagination for queries.
- Performance: Web service must respond each in less than 10 milliseconds
- 4. Use design patterns and good practices: You must use Design patterns learned in this course.
- 5. Testing: You must reach at 80% of Unit Testing coverage in your solution.

You can use the storage you consider the best to provide support to solve this problem. We prefer to use a Relational Database , but if you consider if a NoSQL database fits to comply requirements it's ok.

Optional Activity

In order to have a more comprehensive picture on how Web Services works in a normal environment, you can Dockerize the solution creating containers in an internal network for DB and Web Service and connect both.

Comentario de Nacho a la V1 (Correciones en V2):

• Labels: El usuario puede crear sus propios labels. Solo que hay un typo en el enunciado del ejercicio (Iqual si son predefinidos, no va a estar mal tampoco)

ERD (UML) - Final Thesis Work - V4 "employees" separados de "users" porque (Correción V2) la considero que cada empleado debe tener un columna unico mail coorporativo. En este caso, considero los archivos no tomo en cuenta una mail username pasa employees simularia otra api propia de como links a archivos en dirección de mail, solo el a la tabla empleados, si consideramos el sistema de la nube. En caso que sea username del mismo, ya Employees, porque mensajeria como un servicio aparte. Otra necesario almacenar que al ser un sistema de (tomando como forma es obviar lo anterior y colapsar todos archivos binarios o fotos. una empresa, todos los eiemplo Globant) el los datos en una sola tabla "users". se haran los cambios mails son coorporativos v mail es predefinido pertinentes en el campo comparten el mismo para el empleado. attachment dominio. employees users messages attachments id_employee INT PK id user INT PK id_message INT PK id attachment INT PK username VARCHAR UNO id number INT subject VARCHAR attachment VARCHAR name VARCHAR password VARCHAR body VARCHAR id message INT FK lastname VARCHAR id_employee INT FK user_from INT FK city zip code INT FK is admin BOOLEAN mail username VARCHAR labels_x_user_x_messages labels x_messages recipients id label x user x message id label x message INT PK cities id user INT PFK considero que si is admin id label INT FK id label x user INT FK zip code INT PK es verdadero, el usuario id message INT PFK id message INT FK id message INT FK tendra acceso a visualizar name VARCHAR recipient_type INT PFK los mails del usuario que id state VARCHAR FK desee. labels x user states recipient types id label INT PFK id user INT PFK id state INT PK id_type INT PK labels name VARCHAR name VARCHAR UNO id label INT PK id country INT FK acronvm VARCHAR UNO name VARCHAR (Correción V2) la tabla contendra los countries Cada usuario puede siguientes datos: Primary (Correción V4) crear sus propios id country INT PK Cada usuario podra Receptor(To), Carbon labels. Copy(CC) orBlind Carbon crear sus labels y name VARCHAR UNQ usar los que vienen Copy(BCC) por defecto

Final Thesis work

This is your final thesis to be a certified Glober in Java Web Services.

We need to create a Web Service to allow a company handle a internal messages between employees. Our web service should give support to the following functional requirements :

- 1. User registration: We should have the following data: username, name, last name, identification number, address, zip code city, state, country
- 2. User login: Web service have to have the ability to login a registerd user to send and receive messages.
- 3. Message sending: Web service should allow to send messages to multiple registered users specifiying each recipient as a Primary Receptor (To), Carbon Copy (CC) or Blind Carbon Copy (BCC). User should be logged to sendand receive messages messages should have the following data: Subject, Body and Attachments,
- 4. Message Reception: Web service should allow to receive messages. The user can receive message and catalogue each one with one or more labels.
- 5. Inbox and Sent: Web service will provide support to query messages in inbox and sent.
- 6. Label Filtering: Web service must provide tools to filter messages by label. User can add different labels to each message. Each user can create their own personalized Web Services.

Our Web Service should meet the following Non-Functional requirements:

- 1. RESTFUL: Web service should comply the REST principles.
- 2. Pagination: Web service must allow pagination for queries.
- 3. Performance: Web service must respond each in less than 10 milliseconds
- 4. Use design patterns and good practices: You must use Design patterns learned in this course.
- 5. Testing: You must reach at 80% of Unit Testing coverage in your solution.

You can use the storage you consider the best to provide support to solve this problem. We prefer to use a Relational Database , but if you consider if a NoSQL database fits to comply requirements it's ok.

Optional Activity

In order to have a more comprehensive picture on how Web Services works in a normal environment, you can Dockerize the solution creating containers in an internal network for DB and Web Service and connect both.

Comentario de Nacho a la V1:

• Labels: El usuario puede crear sus propios labels. Solo que hay un typo en el enunciado del ejercicio (Igual si son predefinidos, no va a estar mal tampoco)

ERD (UML) - Final Thesis Work - V5 (Correción V2) la "employees" separados de "users" porque columna considero que cada empleado debe tener un mail username pasa unico mail coorporativo. En este caso, a la tabla employees simularia otra api propia de Employees, porque empleados, si consideramos el sistema de (tomando como recipient types la tabla contendra los mensaieria como un servicio aparte. Otra recipients eiemplo Globant) el forma es obviar lo anterior y colapsar todos siguientes datos: Primary id_type INT PK mail es predefinido los datos en una sola tabla "users". id user INT PFK Receptor(To), Carbon para el empleado. name VARCHAR UNO Copy(CC) orBlind Carbon id message INT PFK Copy(BCC) acronym VARCHAR UNO recipient_type INT PFK employees users messages attachments id_employee INT PK id user INT PK id message INT PK id attachment INT PK username VARCHAR UNO subject VARCHAR id number INT attachment VARCHAR name VARCHAR password VARCHAR body VARCHAR id message INT FK lastname VARCHAR id_employee INT FK user from INT FK city zip code INT FK is admin BOOLEAN labels default x messages mail username VARCHAR id label x message INT PK considero los archivos considero que si is admin id label INT FK como links a archivos en cities es verdadero, el usuario la nube. En caso que sea id message INT FK tendra acceso a visualizar zip code INT PK necesario almacenar los mails del usuario que id user INT FK archivos binarios o fotos. desee. name VARCHAR labels user x messages se haran los cambios pertinentes en el campo id state VARCHAR FK id label x user x message attachment. INT PK labels (Correción V4) id label x user INT FK Cada usuario podra id label INT PK states crear sus labels v id_message **INT FK** name VARCHAR id state INT PK usar los que vienen por defecto is default Boolean name VARCHAR id country INT FK labels user countries id label user INT PK id country INT PK id label INT FK name VARCHAR UNQ id_user INT FK

Final Thesis work

This is your final thesis to be a certified Glober in Java Web Services.

We need to create a Web Service to allow a company handle a internal messages between employees. Our web service should give support to the following functional requirements :

- 1. User registration: We should have the following data: username, name, last name, identification number, address, zip code city, state, country
- 2. User login: Web service have to have the ability to login a registerd user to send and receive messages.
- 3. Message sending: Web service should allow to send messages to multiple registered users specifiying each recipient as a Primary Receptor (To), Carbon Copy (CC) or Blind Carbon Copy (BCC). User should be logged to sendand receive messages messages should have the following data: Subject, Body and Attachments,
- 4. Message Reception: Web service should allow to receive messages. The user can receive message and catalogue each one with one or more labels.
- 5. Inbox and Sent: Web service will provide support to query messages in inbox and sent.
- 6. Label Filtering: Web service must provide tools to filter messages by label. User can add different labels to each message. Each user can create their own personalized Web Services.

Our Web Service should meet the following Non-Functional requirements:

- 1. RESTFUL: Web service should comply the REST principles.
- 2. Pagination: Web service must allow pagination for queries.
- 3. Performance: Web service must respond each in less than 10 milliseconds
- 4. Use design patterns and good practices: You must use Design patterns learned in this course.
- 5. Testing: You must reach at 80% of Unit Testing coverage in your solution.

You can use the storage you consider the best to provide support to solve this problem. We prefer to use a Relational Database , but if you consider if a NoSQL database fits to comply requirements it's ok.

Optional Activity

In order to have a more comprehensive picture on how Web Services works in a normal environment, you can Dockerize the solution creating containers in an internal network for DB and Web Service and connect both.

Comentario de Nacho a la V1:

• Labels: El usuario puede crear sus propios labels. Solo que hay un typo en el enunciado del ejercicio (Igual si son predefinidos, no va a estar mal tampoco)

ERD (UML) - Final Thesis Work - V6 - FINAL "employees" separados de "users" porque considero que cada empleado debe tener un unico mail coorporativo. En este caso, employees simularia otra api propia de recipients empleados, si consideramos el sistema de recipient types la tabla contendra los id recipient INT PK mensaieria como un servicio aparte. Otra forma es obviar lo anterior y colapsar todos siguientes datos: Primary id user INT FK id_type INT PK los datos en una sola tabla "users". Receptor(To), Carbon id message INT FK name VARCHAR UNO Copy(CC) orBlind Carbon Copy(BCC) recipient_type INT FK acronym VARCHAR UNO is deleted by recipient employees users messages id_employee INT PK id user INT PK attachments id message INT PK username VARCHAR UNO id number INT id attachment INT PK name VARCHAR password VARCHAR subject VARCHAR attachment VARCHAR body VARCHAR lastname VARCHAR id_employee INT FK id_message INT FK user from INT FK city_zip_code INT FK is admin BOOLEAN is deleted by user from is enabled BOOLEAN mail username VARCHAR considero los archivos como links a archivos en cities labels_x_messages la nube. En caso que sea id label x message INT PK zip code INT PK necesario almacenar considero que si is admin archivos binarios o fotos. id label INT FK name VARCHAR se haran los cambios es verdadero, el usuario id user INT FK tendra acceso a visualizar pertinentes en el campo id state VARCHAR FK attachment. los mails del usuario que id_message **INT FK** desee. states id state INT PK labels name VARCHAR id label INT PK id country INT FK id user INT FK name VARCHAR countries enabled BOOLEAN id country INT PK

name VARCHAR UNQ

Final Thesis work

This is your final thesis to be a certified Glober in Java Web Services.

We need to create a Web Service to allow a company handle a internal messages between employees. Our web service should give support to the following functional requirements :

- 1. User registration: We should have the following data: username, name, last name, identification number, address, zip code city, state, country
- 2. User login: Web service have to have the ability to login a registerd user to send and receive messages.
- 3. Message sending: Web service should allow to send messages to multiple registered users specifiying each recipient as a Primary Receptor (To), Carbon Copy (CC) or Blind Carbon Copy (BCC). User should be logged to sendand receive messages messages should have the following data: Subject, Body and Attachments.
- 4. Message Reception: Web service should allow to receive messages. The user can receive message and catalogue each one with one or more labels.
- 5. Inbox and Sent: Web service will provide support to query messages in inbox and sent.
- 6. Label Filtering: Web service must provide tools to filter messages by label. User can add different labels to each message. Each user can create their own personalized Web Services.

Our Web Service should meet the following Non-Functional requirements:

- 1. RESTFUL: Web service should comply the REST principles.
- 2. Pagination: Web service must allow pagination for queries.
- 3. Performance: Web service must respond each in less than 10 milliseconds
- 4. Use design patterns and good practices: You must use Design patterns learned in this course.
- 5. Testing: You must reach at 80% of Unit Testing coverage in your solution.

You can use the storage you consider the best to provide support to solve this problem. We prefer to use a Relational Database , but if you consider if a NoSQL database fits to comply requirements it's ok.

Optional Activity

In order to have a more comprehensive picture on how Web Services works in a normal environment, you can Dockerize the solution creating containers in an internal network for DB and Web Service and connect both.