

**DEFENSA HITO 4**

Nombre Completo: Univ. Ricardo David La Madrid Alarcón

Asignatura: PRIGRAMACION III

Carrera: INGENIERÍA DE SISTEMAS

Paralelo: BDA (1)

Docente: Lic. William R. Barra Paredes

fecha: 21/06/2020

**FRAME**

**CLASE FRAME MAIN**

@Component  
public class FrameMain extends JFrame {  
  
 @Autowired  
 private AlphabetPanel alphabetPanel;  
  
  
 public FrameMain() {  
 *this*.setTitle("PROGRA III 2020");  
 *this*.setBounds(300, 200, 800, 600);  
 *this*.setBackground(Color.*blue*);  
 *this*.setLayout(new GridLayout(2, 0));  
  
  
 }  
  
 @PostConstruct  
 public void createPanelsMainFrame() {  
 JPanel container = new JPanel();  
 container.setLayout(new FlowLayout());  
 addPanels(container);  
 *this*.add(container);  
 *this*.setVisible(true);  
  
 }  
  
 public void addPanels(JPanel *container*) {  
 *container*.add(alphabetPanel);  
  
 }  
}

**LISTENER**

**CLASE BUTOON LISTENER**

public class ButtonListener implements *ActionListener* {  
 @Override  
 public void actionPerformed(ActionEvent *e*) {  
 JButton currentButton = (JButton) *e*.getSource();  
 JOptionPane.*showMessageDialog*(null, "Button Pressed -> " + currentButton.getText());  
 }  
}

**PANELS**

**CLASE ALPHABET PANEL**

@Component  
public class AlphabetPanel extends JPanel {  
 JLabel word,lengu,resul;  
 JTextField t1,t2,t3;  
 JButton b1,b2;  
 @Autowired  
 private AlphabetService alphabetService;  
  
  
 public AlphabetPanel() {  
 System.*setProperty*("butBackColor", "#C1ECF1");  
 System.*setProperty*("textColor", "#0B0BF6");  
 *this*.setPreferredSize(new Dimension(600, 300));  
 *this*.setBackground(Color.*darkGray*);  
 *this*.setLayout(new GridLayout(5, 0));  
  
  
  
  
 t1 = new JTextField();  
 t1.setPreferredSize(new Dimension(90,30));  
 t2 = new JTextField();  
 t2.setPreferredSize(new Dimension(90,30));  
  
 t3 = new JTextField();  
 t3.setPreferredSize(new Dimension(90,30));  
  
 b1 = new JButton("TRANSLATE");  
 b1.setPreferredSize(new Dimension(120,30));  
  
 b2 = new JButton("CLEAN");  
 b2.setPreferredSize(new Dimension(120,30));  
 }  
  
 @PostConstruct  
 public void createButtonsLetters() {  
 *List*<AlphabetModel> firstRow = alphabetService.getAllLettersFirst();  
 String[] titleAlphabet = firstRow.get(0).getLetter().split(",");  
 JPanel panelQ\_P = *this*.createPanelButton(titleAlphabet);  
 *this*.add(panelQ\_P);  
  
 *List*<AlphabetModel> secondRow = alphabetService.getAllLettersSecond();  
 String[] titleAlphabet1 = secondRow.get(0).getLetter().split(",");  
 JPanel panelA\_L = *this*.createPanelButton(titleAlphabet1);  
 *this*.add(panelA\_L);  
  
 *List*<AlphabetModel> threeRow = alphabetService.getAllLettersThree();  
 String[] titleAlphabet2 = threeRow.get(0).getLetter().split(",");  
 JPanel panelZ\_M = *this*.createPanelButton(titleAlphabet2);  
 *this*.add(panelZ\_M);  
  
 JPanel panel1 = *this*.createPanel1();  
 *this*.add(panel1);  
 JPanel panel2 = *this*.createPanel2();  
 *this*.add(panel2);  
 }  
  
 public JPanel createPanelButton(String[] *titleAlphabet*) {  
 JPanel mainPanel = new JPanel();  
 mainPanel.setLayout(new FlowLayout());  
 ButtonListener listener = new ButtonListener();  
  
 for (String title : *titleAlphabet*) {  
 JButton button = new JButton(title);  
 button.setPreferredSize(new Dimension(55, 40));  
 button.addActionListener(listener);  
 button.setBackground(Color.*getColor*("butBackColor"));  
 button.setForeground(Color.*getColor*("textColor"));  
 button.setBorder(BorderFactory.*createEmptyBorder*());  
 *// button.setFont(Util.FONT\_TEXT);* mainPanel.add(button);  
 }  
 return mainPanel;  
  
 }  
  
@PostConstruct  
 public JPanel createPanel1() {  
 JPanel mainPanel1 = new JPanel();  
 mainPanel1.setLayout(new FlowLayout());  
 word = new JLabel("WORD");  
 word.setFont(new Font("Yu Gothic UI Semilight", Font.*PLAIN*, 15));  
 word.setPreferredSize(new Dimension(80,30));  
 mainPanel1.add(word);  
 mainPanel1.add(t1);  
 lengu = new JLabel("LENGUAJE");  
 lengu.setFont(new Font("Yu Gothic UI Semilight", Font.*PLAIN*, 15));  
 lengu.setPreferredSize(new Dimension(80,30));  
 mainPanel1.add(lengu);  
 mainPanel1.add(t2);  
 resul = new JLabel("RESULTADO");  
 resul.setFont(new Font("Yu Gothic UI Semilight", Font.*PLAIN*, 15));  
 resul.setPreferredSize(new Dimension(80,30));  
 mainPanel1.add(resul);  
 mainPanel1.add(t3);  
  
 return mainPanel1;  
 }  
  
 @PostConstruct  
 public JPanel createPanel2() {  
 JPanel mainPanel1 = new JPanel();  
 mainPanel1.setLayout(new FlowLayout());  
  
 mainPanel1.add(b1);  
 mainPanel1.add(b2);  
  
  
 return mainPanel1;  
 }  
}

**MODEL**

**CLASE ALPHABET MODEL**

@Entity  
@Table (name = "Alphabet")  
public class AlphabetModel {  
 @Id  
 @GeneratedValue(strategy = *GenerationType*.*AUTO*)  
 private Integer Id;  
  
 @Column(name = "letter", length = 200, nullable = false)  
 private String letter;  
  
 @Column(name = "typeR", length = 100, nullable = false)  
 private String typeR;  
  
 public AlphabetModel() {  
  
 }  
  
 public AlphabetModel(String *letter*, String *typeR*) {  
 *this*.letter = *letter*;  
 *this*.typeR = *typeR*;  
 }  
  
 public String getTypeR() {  
 return typeR;  
 }  
  
 public void setTypeR(String *typeR*) {  
 *this*.typeR = *typeR*;  
 }  
  
 public Integer getId() {  
 return Id;  
 }  
  
 public void setId(Integer *id*) {  
 Id = *id*;  
 }  
  
 public String getLetter() {  
 return letter;  
 }  
  
 public void setLetter(String *letter*) {  
 *this*.letter = *letter*;  
 }  
}

**CLASE DICTIONARY MODEL**

@Entity  
@Table(name = "Dictionary")  
public class DictionaryModel {  
 @javax.persistence.Id  
 @GeneratedValue(strategy = *GenerationType*.*AUTO*)  
 private Integer Id;  
  
 @Column(name = "english", length = 200, nullable = false)  
 private String english;  
  
 @Column(name = "portugues", length = 100, nullable = false)  
 private String portugues;  
  
 @Column(name = "word", length = 100, nullable = false)  
 private String word;  
  
 public DictionaryModel() {  
  
 }  
  
 public DictionaryModel(String *english*, String *portugues*, String *word*) {  
 *this*.english = *english*;  
 *this*.portugues = *portugues*;  
 *this*.word = *word*;  
 }  
  
 public Integer getId() {  
 return Id;  
 }  
  
 public void setId(Integer *id*) {  
 Id = *id*;  
 }  
  
 public String getEnglish() {  
 return english;  
 }  
  
 public void setEnglish(String *english*) {  
 *this*.english = *english*;  
 }  
  
 public String getPortugues() {  
 return portugues;  
 }  
  
 public void setPortugues(String *portugues*) {  
 *this*.portugues = *portugues*;  
 }  
  
 public String getWord() {  
 return word;  
 }  
  
 public void setWord(String *word*) {  
 *this*.word = *word*;  
 }  
}

**REPO**

**INTERFACE ALPHABET REPOSITORY**

public interface *AlphabetRepository* extends *JpaRepository*<AlphabetModel, Integer> {  
  
 @Query(value="SELECT \* FROM alphabet WHERE typer ='first';", nativeQuery=true)  
 public *List*<AlphabetModel> getFirstRow();  
  
 @Query(value="SELECT \* FROM alphabet WHERE typer ='second';", nativeQuery=true)  
 public *List*<AlphabetModel> getSecondRow();  
  
 @Query(value="SELECT \* FROM alphabet WHERE typer ='three';", nativeQuery=true)  
 public *List*<AlphabetModel> getThreeRow();  
}

**INTERFACE DICTIONARY REPOSITORY**

public interface *DictionaryRepository* extends *JpaRepository*<DictionaryModel,Integer> {}

**SERVICE**

**CLASE ALPHABET SERVICE**

@Service  
public class AlphabetService implements *AlphabetServiceInterface* {  
 private static final String *Q\_P* = "Q,W,E,R,T,Y,U,I,O,P";  
 private static final String *A\_L* = "A,S,D,F,G,H,J,K,L";  
 private static final String *Z\_M* = "Z,X,C,V,B,N,M";  
 @Autowired  
 private *AlphabetRepository* alphabetRepository;  
  
 @Override  
 public void saveData() {  
 if (alphabetRepository.count() == 0) {  
 alphabetRepository.save(new AlphabetModel(*Q\_P*, "first"));  
 alphabetRepository.save(new AlphabetModel(*A\_L*, "second"));  
 alphabetRepository.save(new AlphabetModel(*Z\_M*, "three"));  
 }  
 }  
  
 @Override  
 public *List*<AlphabetModel> getAllLettersFirst() {  
 return alphabetRepository.getFirstRow();  
 }  
  
 @Override  
 public *List*<AlphabetModel> getAllLettersSecond() {  
 return alphabetRepository.getSecondRow();  
 }  
  
 @Override  
 public *List*<AlphabetModel> getAllLettersThree() {  
 return alphabetRepository.getThreeRow();  
 }  
}

**INTERFACE ALPHABETSERVICE**

public interface *AlphabetServiceInterface* {  
 public void saveData();  
 public *List*<AlphabetModel> getAllLettersFirst();  
 public *List*<AlphabetModel> getAllLettersSecond();  
 public *List*<AlphabetModel> getAllLettersThree();  
}

**CLASE DICTIONARY SERVICE**

@Service  
public class DictionaryService implements *DictionaryServiceInterface*{  
 @Autowired  
 private *DictionaryRepository* dictionaryRepository;  
  
 @Override  
 public void saveData() {  
 if (dictionaryRepository.count() == 0) {  
  
  
 dictionaryRepository.save(new DictionaryModel("Monday","Segunda-Feira","Lunes"));  
 dictionaryRepository.save(new DictionaryModel("Tuesday","Terca-Feira","Martes"));  
 dictionaryRepository.save(new DictionaryModel("Wednesday","Quarta-Feira","Miercoles"));  
 dictionaryRepository.save(new DictionaryModel("Thursday","Quinta-Feira","Jueves"));  
 dictionaryRepository.save(new DictionaryModel("Friday","Sexta-Feira","Viernes"));  
 dictionaryRepository.save(new DictionaryModel("Saturday","Sabado","Sabado"));  
 dictionaryRepository.save(new DictionaryModel("Sunday","Domingo","Domingo"));  
  
 }  
 }  
  
 @Override  
 public *List*<DictionaryModel> getAllEnglis() {  
 return null;  
 }  
  
 @Override  
 public *List*<DictionaryModel> getAllPortugues() {  
 return null;  
 }  
  
 @Override  
 public *List*<DictionaryModel> getAllWord() {  
 return null;  
 }  
}

**INTERFACE DICTIONARY SERVICE**

public interface *DictionaryServiceInterface* {  
 public void saveData();  
 public *List*<DictionaryModel> getAllEnglis();  
 public *List*<DictionaryModel> getAllPortugues();  
 public *List*<DictionaryModel> getAllWord();  
}