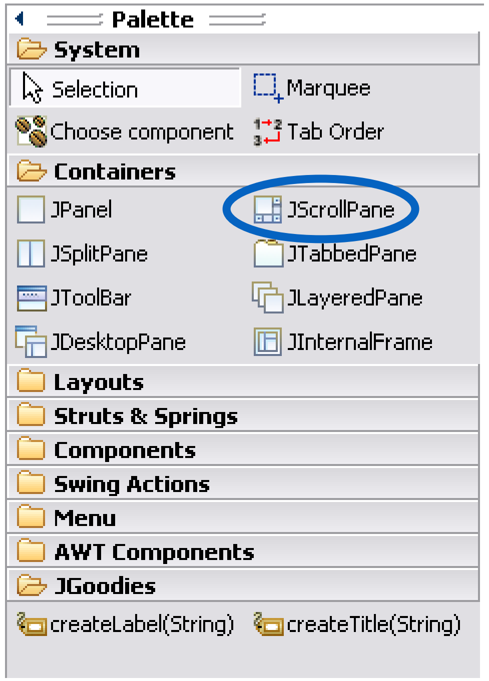
Theory pills
Lab 4

Main types of containers

- Frame (JFrame)
- Dialog (JDialog)
- Panel (JPanel)
- **Scroll Panel (JScrollPane)**
- Tabbed panel (JTabbedPane)
- Tool bar (JToolBar)

Scroll Pane (JScrollPane)



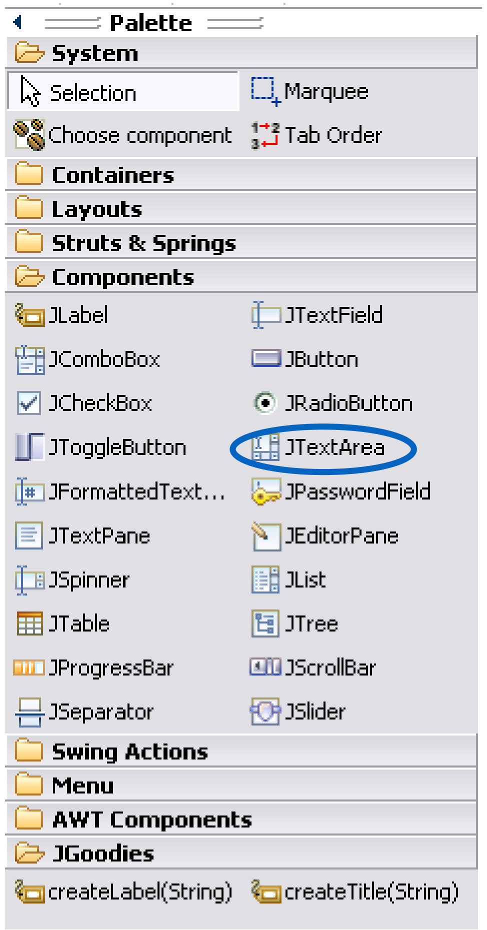
- Specialized container that provides scroll bars. The user can shift the visible part of the content of the window.

- We can force the scroll bars to be visible ALWAYS or only when there are needed (AS_NEEDED).
  - Properties: verticalScrollBarPolicy y horizontalScrollBarPolicy
  - Generally, only AS_NEEDED.

Text components

- The allow the user to see and edit text
- Some components…
  - *JLabel*
  - *JTextField*
  - *JPasswordField*
  - ***JTextArea***

Text Area (JTextArea)



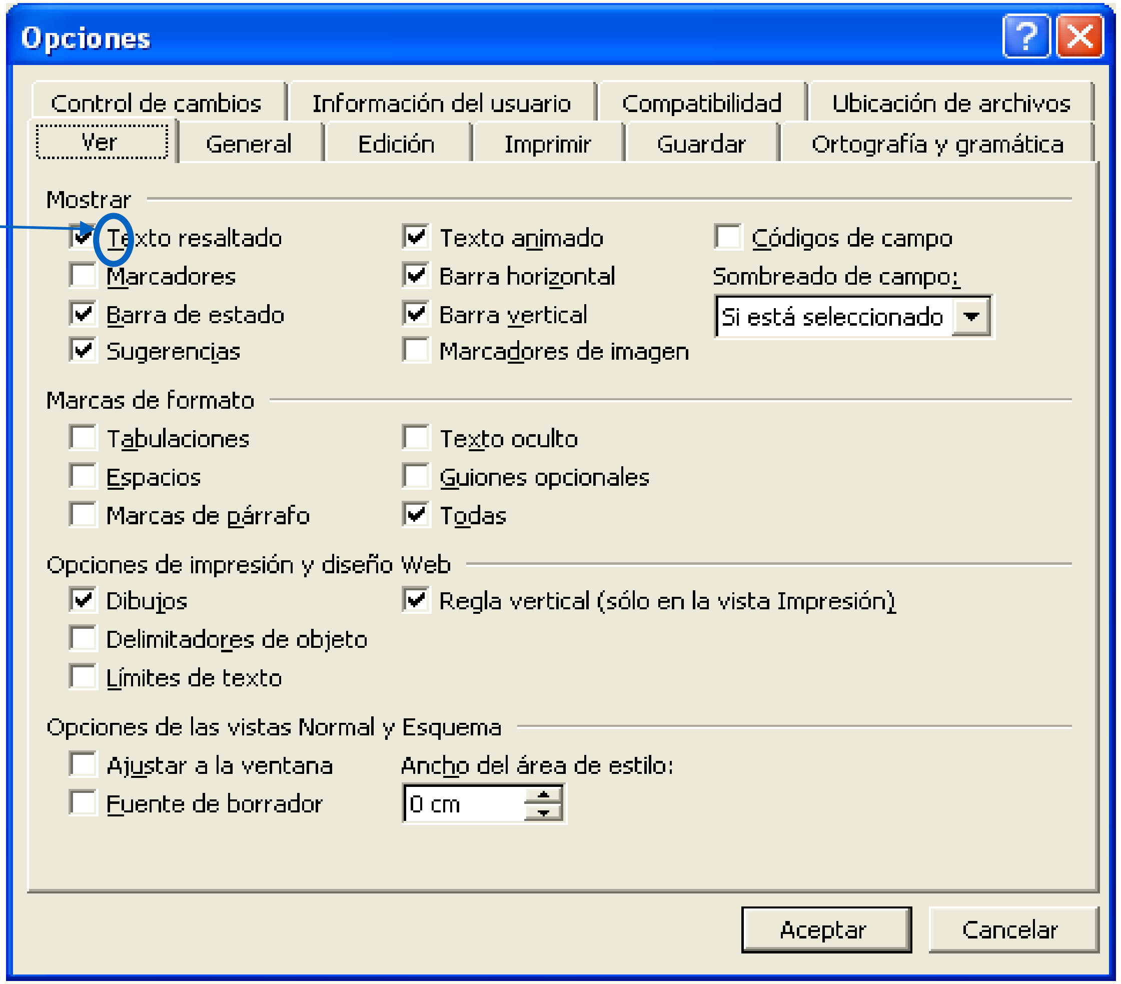
- Provides a space where users can see, key and edit multiple lines of text.
- It has founts, size and simple style.
- If we want scroll bars it must be included inside a Scroll Pane
- Properties
    - lineWrap->True
    - wrapStyleWord-> True
- Methods
    - *setText* //
    - *append* //
    - *insert* //

Keyboard operations

- There must always be an alternative interaction method to the mouse in order to:
  - Help users that are used to different environments
  - Help handicapped users
  - Prevent mouse failures.
- Good practices:
  - Use of mnemonics
  - Shortcuts
  - Keyboard navigation and activation

Mnemonics(I)

- It is an underlined character that is usually present in titles, menu options, button text, etc.
- The underlined character evidences the way the user can active the command::
    - <ALT> + <underlined character>



Mostrar

☑ Texto resaltado
□ Marcadores
☑ Barra de estado
☑ Sugerencias
☑ Texto animado
☑ Barra horizontal
☑ Barra vertical
□ Marcadores de imagen
□ Códigos de campo
Sombreado de campo: Si está seleccionado

Marcas de formato

□ Tabulaciones
□ Espacios
□ Marcas de párrafo
□ Texto oculto
□ Guiones opcionales
☑ Todas

Opciones de impresión y diseño Web

☑ Dibujos
□ Delimitadores de objeto
□ Límites de texto
☑ Regla vertical (sólo en la vista Impresión)

Opciones de las vistas Normal y Esquema

□ Ajustar a la ventana
□ Fuente de borrador
Ancho del área de estilo: 0 cm

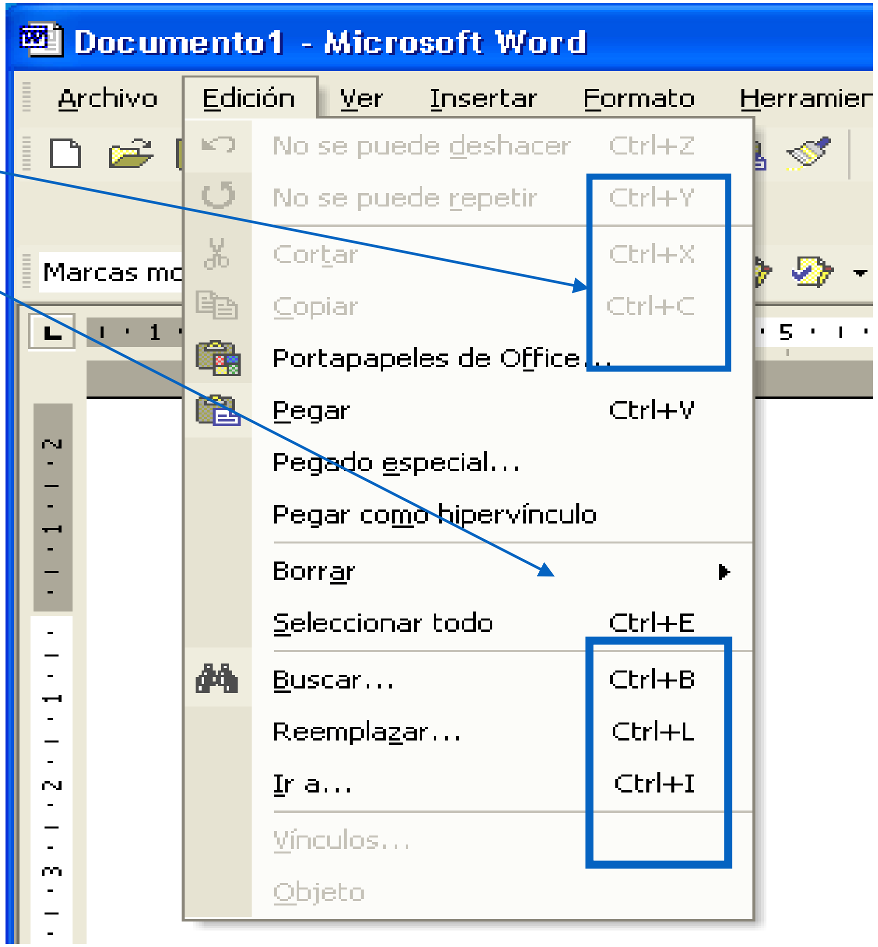
Mnemonics (II)

- Some properties and methods related with mnemonics…
    - *mnemonic,* for those components that admit it directly (buttons, checkboxes, radio-buttons, titles, menu items, etc.).
    - *displayedMnemonic*, for labels that complements components that do not admit mnemonics (comboboxes, textfields, etc.).
- Mnemonics must apply the following rules
    - Avoid conflicts.
    - Choose the first character of the menu
    - If the first character come into conflict with any other, we must choose a prominent consonant (T,X,Z,P,G,K…)

        Ex: T for *Cu**t***
    - If both the first character and the promient consonant rule come into conflict, then we must choose a prominent vowel (e,o,a)

Shortcuts

- Keys sequence that activates a menu option
- Key combination is formed by:
  - *Control* modifier (and optionally, another like *Shift*)
  - A character or a function key (F1,F2, etc)
- They must be consistent with the usual shortcuts in the platform.
  - Example:
    - Copy Ctrl+C
    - Print Ctrl+P
    - …



Keyboard focus

- Also named “input focus”
- It points out the active windows and/or the component where the next keystroke will affect.
- The first time a window is opened the focus must point to the component that the user will interact with first.
  - Generally, the closer to the upper-left corner (Western cultures!)
  - This is especially important for people that interact only through the keyboard (example, users with motor disabilities).

Keyboard navigation and activation

- Keyboard navigation and activation allow us to move the focus along the components of the interface using the keyboard.
    - Tab-> moves the focus to the next component
    - Shitf-Tab-> moves the focus to the previous component

- We must ensure that every option is available through the keyboard!