CONTROLLER

- Controller()

Controller

- can_dao= new CandidateDAO();
- v_principal = new VMain();
- ActionListener
- ActionCommand

- Controller()
- start(): void
- actionPerfomed(ActionEvent e): void

Apl Main

Controller c = new Controller();

- Main()
- c.start()

MODEL
CandidateDTO

- name: String
- last_name: String
- id: int
- age: int
- rank: String

+ CandidatoDTO(String name, String last_name, int id, int age, String rank)

+Getters
+Setters

+ToString(): String

OperationsDAO
(Interface)

+ register (Object o): abstract void

CandidateDAO

- ArrayList<CandidateDTO> list;
- fc: FileController

+ register (Object o): void
+ modify (String name, String last_name, int id, int age, String rank, int id_change): boolean
+ create (String name, String last_name, int id, int age, String rank): CandidateDTO
+ showID(int id): CandidateDTO
+ idlist (): String
+ erase(int id): boolean

MODEL.PERSISTANCE
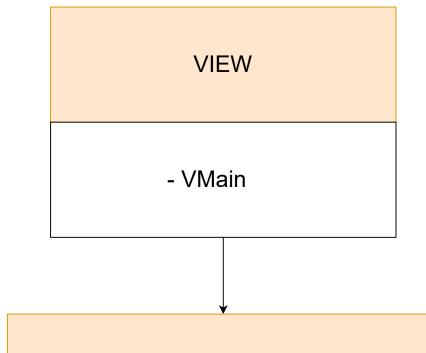## FileController

FileController

FileController

- FileOutputStream(archivo_salida)
- ObjectOutputStream(escritor)
- FileInputStream(archivo_entrada)
- ObjectInputStream(lector)
- save.ini - saves.ini

+ initSalida(String nombreclase): void
+ escribirSerializado(Object o): void
+ cerrarSerializado(): void
+ initEntrada(String nombreclase): void
+ leerSerializado(): Object
+ cerrarLector(): void

VIEW

- VMain

VMain

- JPanel (register,showall,showids)
- JLabel (t_tittle, t_name, t_last_name, t_id, t_age, t_rank, t_action, t_register, t_showall, t_list, t_delete. fondo)
- JTextField (name_candidate, last_name, id, age, rank, id_actions)
- JButton (bregister, bshowall, bdelete, bmodify, blist)
- JTextArea (showcase)

+ eraseData()
+ getters()
+ setters()