**Diagrama de Classes - Sistema de Reservas**

**ENUMS**

**UserType**

* ADMIN
* USER
* getDescription()

**ReservationStatus**

* PENDING
* CONFIRMED
* CANCELED
* CONCLUDED
* getDescription()

**CLASSES MODELO**

**User**

* id: int
* name: String
* email: String
* password: String
* type: UserType
* creationDate: LocalDateTime
* getId(), setId(), getName(), setName()
* getEmail(), setEmail(), getPassword(), setPassword()
* getType(), setType(), getCreationDate(), setCreationDate()

**Space (abstrata)**

* id: int
* name: String
* localization: String
* capacity: int
* availableHours: String
* creationDate: LocalDateTime
* getId(), setId(), getName(), setName()
* getLocalization(), setLocalization()
* getCapacity(), setCapacity()
* getAvailableHours(), setAvailableHours()
* getType() (abstrato)
* getSpecifiedDescription() (abstrato)

**Classroom extends Space**

* numOfBoards: int
* numOfDataShows: int
* getNumOfBoards(), setNumOfBoards()
* getNumOfDataShows(), setNumOfDataShows()

**Laboratory extends Space**

* numOfComputers: int
* labType: String
* getNumOfComputers(), setNumOfComputers()
* getLabType(), setLabType()

**MeetingRooms extends Space**

* numOfProjectors: int
* numOfVideoConferences: int
* getNumOfProjectors(), setNumOfProjectors()
* getNumOfVideoConferences(), setNumOfVideoConferences()

**Auditorium extends Space**

* numOfSoundSystems: int
* numOfStages: int
* getNumOfSoundSystems(), setNumOfSoundSystems()
* getNumOfStages(), setNumOfStages()

**SportsField extends Space**

* numOfGoals: int
* numOfNets: int
* numOfHoops: int
* getNumOfGoals(), setNumOfGoals()
* getNumOfNets(), setNumOfNets()
* getNumOfHoops(), setNumOfHoops()

**Reservation**

* id: int
* userId: int
* spaceId: int
* status: ReservationStatus
* startDateTime: LocalDateTime
* endDateTime: LocalDateTime
* description: String
* creationDate: LocalDateTime
* space: Space
* getId(), setId(), getUserId(), setUserId()
* getSpaceId(), setSpaceId(), getStatus(), setStatus()
* getStartDateTime(), setStartDateTime()
* getEndDateTime(), setEndDateTime()
* getDescription(), setDescription()
* getSpace(), setSpace()

**INTERFACE**

**DataAccessObject<T>**

* insert(T): boolean
* update(T): boolean
* delete(T): boolean
* findById(int): T
* listAll(): List<T>

**CLASSES DAO**

**UserDao implements DataAccessObject<User>**

* insert(), update(), delete(), findById(), listAll()
* findByEmail(String): User
* validateLogin(String, String): boolean

**SpaceDao implements DataAccessObject<Space>**

* insert(), update(), delete(), findById(), listAll()
* findByType(String): List<Space>

**ReservationDao implements DataAccessObject<Reservation>**

* insert(), update(), delete(), findById(), listAll()
* listByUserId(int): List<Reservation>

**CONTROLLERS**

**UserController**

* userDao: UserDao
* createUser(): boolean
* updateUser(): boolean
* deleteUser(): boolean
* login(): User
* getUserById(): User
* getUserByEmail(): User
* listAllUsers(): List<User>
* listUsersByType(): List<User>
* authenticateUser(): boolean
* changePassword(): boolean
* emailExists(): boolean
* isAdmin(): boolean
* createAdmin(): boolean

**SpaceController**

* spaceDao: SpaceDao
* createClassroom(): boolean
* createLaboratory(): boolean
* createMeetingRoom(): boolean
* createSportsField(): boolean
* createAuditorium(): boolean
* updateSpace(): boolean
* deleteSpace(): boolean
* getSpaceById(): Space
* listAllSpaces(): List<Space>
* listSpacesByType(): List<Space>

**ReservationController**

* reservationDao: ReservationDao
* createReservation(): boolean
* updateReservation(): boolean
* cancelReservation(): boolean
* confirmReservation(): boolean
* concludeReservation(): boolean
* changeReservationStatus(): boolean
* deleteReservation(): boolean
* getReservationById(): Reservation
* listAllReservations(): List<Reservation>
* hasTimeConflict(): boolean
* validateReservation(): boolean
* getReservationsByUserId(): List<Reservation>

**CLASSES UTILITÁRIAS**

**connectionDB**

* URL: String
* USER: String
* PASSWORD: String
* getConnection(): Connection
* endConnection(Connection): void
* testConnection(): boolean

**Logger**

* FILE\_PATH: String
* log(String): void
* logIfSuccess(boolean, String): void

## CLASSES DE INTERFACE GRÁFICA

**LoginFrame extends JFrame**

* emailField: JTextField
* passwordField: JPasswordField
* userController: UserController

**Métodos:**

* setupUI(): void
* createStyledButton(String, Color): JButton
* login(): void
* openRegisterFrame(): void
* showLoginFrame(): void
* showError(String): void

**RegisterFrame extends JFrame**

* nameField: JTextField
* emailField: JTextField
* passwordField: JPasswordField
* userTypeCombo: JComboBox<String>
* adminKeyField: JTextField
* adminKeyLabel: JLabel
* userController: UserController
* loginFrame: LoginFrame
* setupUI(): void
* createFieldPanel(String, JComponent): JPanel
* createTextField(): JTextField
* createPasswordField(): JPasswordField
* createStyledButton(String, Color): JButton
* createAccount(): void
* goBackToLogin(): void
* showError(String): void

**MenuFrame extends JFrame**

* user: User
* configureWindow(): void
* showMenu(): void
* addButton(String, ActionListener): void
* openScreen(JFrame): void

**ReservationsSpaceScreen extends JFrame**

* reservationController: ReservationController
* spaceController: SpaceController
* loggedUserId: int
* spaceCombo: JComboBox<String>
* dateField: JTextField
* startTimeField: JTextField
* endTimeField: JTextField
* purposeField: JTextField
* setupUI(): void
* loadSpaces(JComboBox<String>): void
* verificarDisponibilidade(): void
* fazerReserva(): void
* camposValidos(): boolean
* extractSpaceId(String): int
* parseDateTime(String, String): LocalDateTime
* checkAvailability(int, LocalDateTime, LocalDateTime): boolean
* clearFields(JTextField...): void

**ViewReservationsScreen extends JFrame**

**Métodos:**

* ViewReservationsScreen(User): constructor

**CancelReservationsScreen extends JFrame**

* user: User
* reservationController: ReservationController
* spaceController: SpaceController
* reservationTable: JTable
* tableModel: DefaultTableModel
* cancelReservationButton: JButton
* initializeComponents(): void
* setupLayout(): void
* setupListeners(): void
* loadReservations(): void
* cancelSelectedReservation(): void
* showCancelReservationsScreen(User): void

**RegisterSpaceScreen extends JFrame**

* user: User
* spaceController: SpaceController
* nameField: JTextField
* localizationField: JTextField
* capacityField: JTextField
* availableHoursField: JTextField
* spaceTypeComboBox: JComboBox<String>
* specificFieldsPanel: JPanel
* cardLayout: CardLayout
* numBoardsField: JTextField
* numDataShowsField: JTextField
* numComputersField: JTextField
* labTypeField: JTextField
* numProjectorsField: JTextField
* numVideoConferencesField: JTextField
* numGoalsField: JTextField
* numNetsField: JTextField
* numHoopsField: JTextField
* numSoundSystemsField: JTextField
* numStagesField: JTextField
* registerButton: JButton
* clearButton: JButton
* cancelButton: JButton
* initializeComponents(): void
* initializeSpecificFields(): void
* setupSpecificFieldsPanels(): void
* createClassroomPanel(): JPanel
* createLaboratoryPanel(): JPanel
* createMeetingRoomPanel(): JPanel
* createSportsFieldPanel(): JPanel
* createAuditoriumPanel(): JPanel
* setupLayout(): void
* createGeneralFieldsPanel(): JPanel
* createButtonPanel(): JPanel
* setupEventListeners(): void
* registerSpace(): void
* validateGeneralFields(): boolean
* validateSpecificFields(String): boolean
* clearFields(): void
* showMessage(String, String, int): void
* showRegisterSpace(User): void

**DeleteSpaceScreen extends JFrame**

* user: User
* spaceController: SpaceController
* spaceTable: JTable
* tableModel: DefaultTableModel
* tableSorter: TableRowSorter<DefaultTableModel>
* searchField: JTextField
* typeFilterComboBox: JComboBox<String>
* searchButton: JButton
* clearFilterButton: JButton
* refreshButton: JButton
* deleteButton: JButton
* viewDetailsButton: JButton
* selectedSpace: Space
* initializeComponents(): void
* setupLayout(): void
* createFilterPanel(): JPanel
* createButtonPanel(): JPanel
* setupEventListeners(): void
* loadSpaces(): void
* applyFilters(): void
* mapLabelToInternalType(String): String
* clearFilters(): void
* deleteSpace(): void
* showSpaceDetails(): void
* showMessage(String, String, int): void
* showDeleteSpace(User): void

**SpaceListingScreen extends JFrame**

* spaceController: SpaceController
* spaceTable: JTable
* tableModel: DefaultTableModel
* tableSorter: TableRowSorter<DefaultTableModel>
* searchField: JTextField
* typeFilterComboBox: JComboBox<String>
* searchButton: JButton
* clearFilterButton: JButton
* refreshButton: JButton
* viewDetailsButton: JButton
* selectedSpace: Space
* initializeComponents(): void
* setupLayout(): void
* createFilterPanel(): JPanel
* createButtonPanel(): JPanel
* setupEventListeners(): void
* loadSpaces(): void
* applyFilters(): void
* mapLabelToInternalType(String): String
* clearFilters(): void
* showSpaceDetails(): void
* showSpaceListing(): void

**UserManagementScreen extends JFrame**

* userController: UserController
* userTable: JTable
* tableModel: DefaultTableModel
* nameField: JTextField
* emailField: JTextField
* passwordField: JTextField
* typeComboBox: JComboBox<UserType>
* addButton: JButton
* updateButton: JButton
* deleteButton: JButton
* refreshButton: JButton
* selectedUser: User
* initializeComponents(): void
* setupLayout(): void
* createFormPanel(): JPanel
* createButtonPanel(): JPanel
* setupEventListeners(): void
* loadUsers(): void
* loadUserToForm(int): void
* addUser(): void
* updateUser(): void
* deleteUser(): void
* validateFields(): boolean
* clearFields(): void
* showMessage(String, String, int): void
* showUserManagement(): void

**ReservationManagementScreen extends JFrame**

* reservationController: ReservationController
* spaceController: SpaceController
* userController: UserController
* reservationTable: JTable
* tableModel: DefaultTableModel
* statusFilterCombo: JComboBox<ReservationStatus>
* spaceFilterCombo: JComboBox<Space>
* userFilterCombo: JComboBox<User>
* filterButton: JButton
* clearFilterButton: JButton
* cancelButton: JButton
* confirmButton: JButton
* concludeButton: JButton
* deleteButton: JButton
* refreshButton: JButton
* selectedReservation: Reservation
* statusLabel: JLabel
* userCache: Map<Integer, User>
* spaceCache: Map<Integer, Space>
* cachedReservations: List<Reservation>
* dateFormatter: DateTimeFormatter
* showLoadingMessage(): void
* loadInitialDataAsync(): void
* loadCacheData(): boolean
* finishInitialization(): void
* showErrorAndClose(): void
* initializeComponents(): void
* setupButtonColors(): void
* setupTable(): void
* setupMainLayout(): void
* setupLayout(): void
* createFilterComboPanel(String, JComboBox<?>): JPanel
* setupEventListeners(): void
* findReservationInCache(int): Reservation
* loadFiltersFromCache(): void
* loadReservationsFromCache(): void
* changeReservationStatus(ReservationStatus): void
* getActionName(ReservationStatus): String
* deleteReservation(): void
* updateActionButtons(boolean): void
* showMessage(String, String, int): void

**RELACIONAMENTOS**

**User:**

* possui UserType

**Reservation:**

* possui ReservationStatus
* referencia Space e User

**Classroom, Laboratory, MeetingRooms, Auditorium e SportsField:**

* herdam de Space

**UserController:**

* usa UserDao

**SpaceController:**

* usa SpaceDao
* ReservationController
* usa ReservationDao

**Todos os Controllers:**

* usam Logger

**Todos os DAOs:**

* usam connectionDB

**LoginFrame:**

* Usa UserController
* Cria RegisterFrame
* Cria MenuFrame

**RegisterFrame:**

* Usa UserController
* Referencia LoginFrame

**MenuFrame:**

* Possui User
* Cria todas as telas de funcionalidades

**ReservationsSpaceScreen:**

* Usa ReservationController
* Usa SpaceController

**CancelReservationsScreen:**

* Possui User
* Usa ReservationController
* Usa SpaceController

**RegisterSpaceScreen:**

* Possui User
* Usa SpaceController

**DeleteSpaceScreen:**

* Possui User
* Usa SpaceController
* Referencia Space

**SpaceListingScreen:**

* Usa SpaceController
* Referencia Space

**UserManagementScreen:**

* Usa UserController
* Referencia User

**ReservationManagementScreen:**

* Usa ReservationController
* Usa SpaceController
* Usa UserController
* Referencia Reservation, User, Space
* Usa cache para otimização