

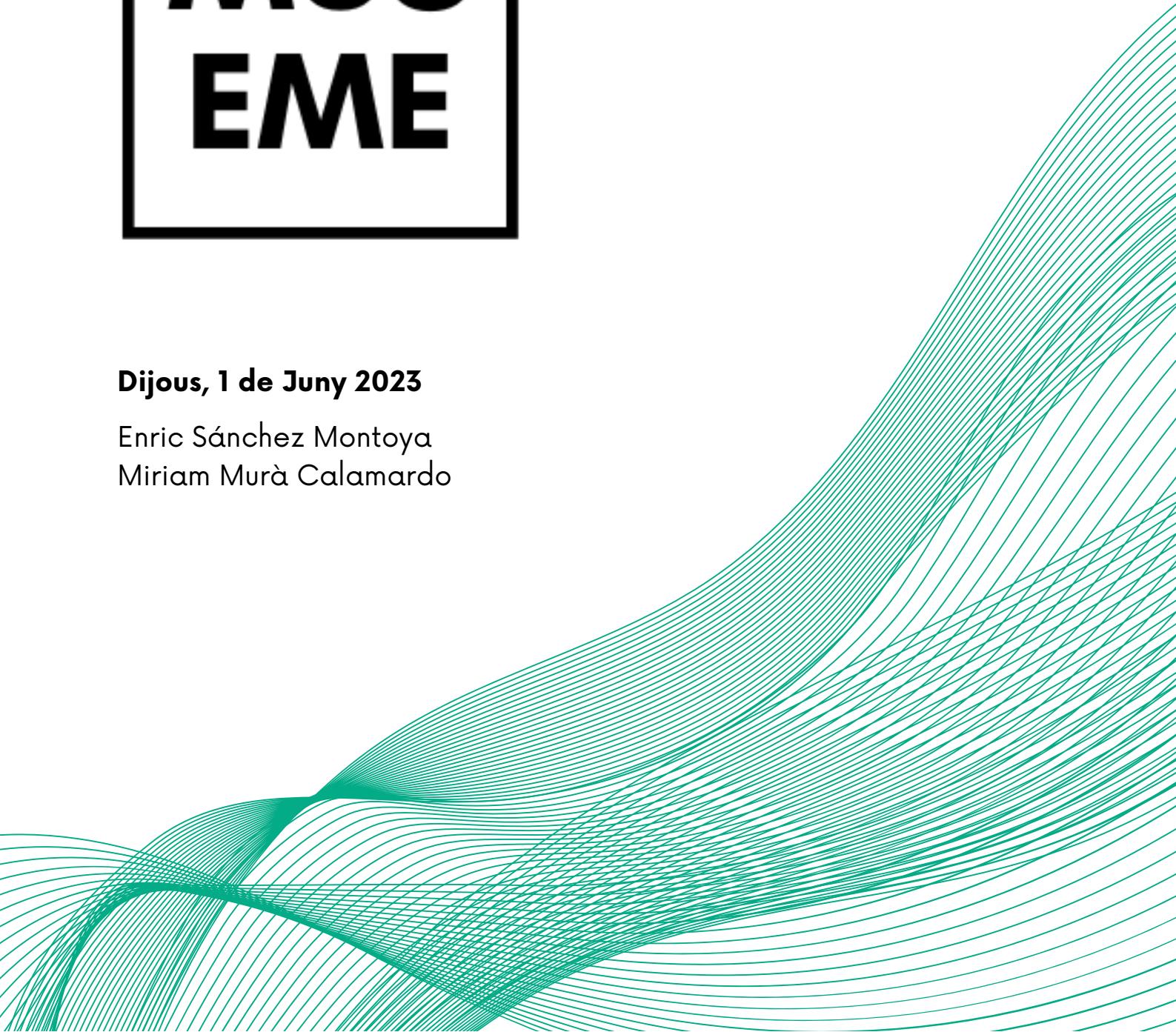
PROJECTE FINAL 1R DAM

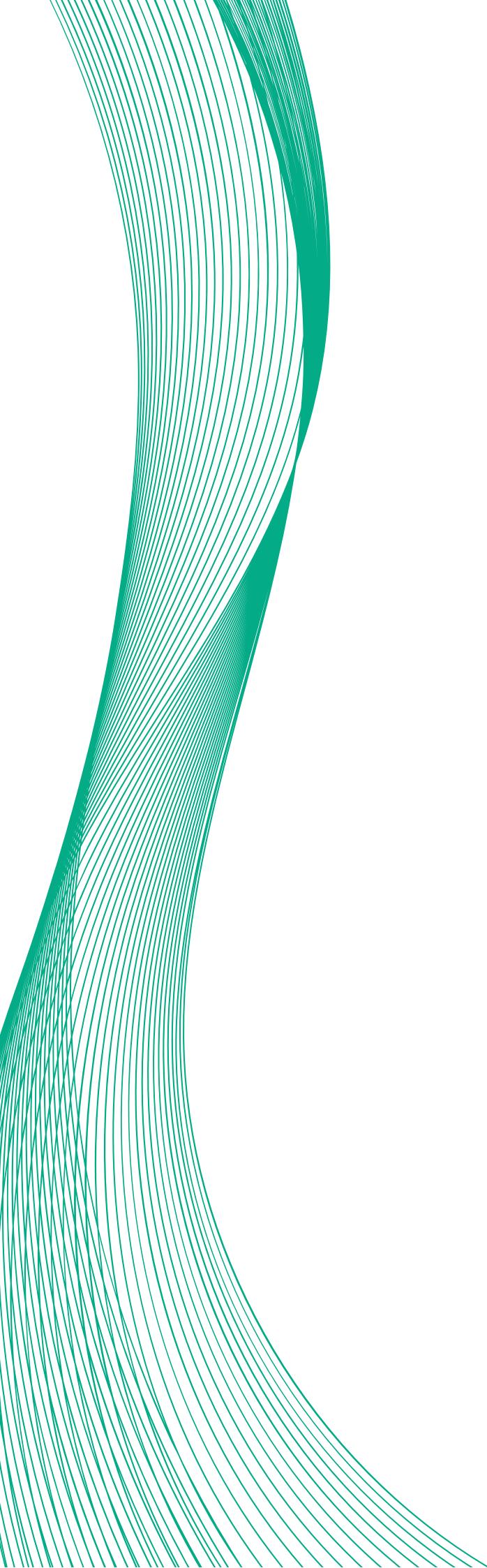


**MUS
EME**

Dijous, 1 de Juny 2023

Enric Sánchez Montoya
Miriam Murà Calamardo





ÍNDEX

01

ANÁLISIS FUNCIONAL

02

DIAGRAMA FUNCIONAL
USUARI

03

MOCKUPS

04

DIAGRAMA ENTITAT-RELACIÓ

05

DIAGRAMA UML
INICIAL

06

DIAGRAMA UML
GENERAT PER L'IDE

07

KANBANFLOW

ANÁLISIS FUNCIONAL

01 Pantalla de Presentació:

Pantalla en blanc/negre amb el logotip al mig (transició).

02 Pantalla de registre i pantalla d'inici de sessió:

1. Inici de sessió::
 - Camp del correu electrònic
 - Camp de la contrasenya (comprovar que la contrasenya sigui segura)
2. Registre:
 - Logotip de l'aplicació
 - Camp del nom
 - Camp del cognom
 - Camp del correu electrònic
 - Camp de la contrasenya (comprovar que la contrasenya sigui segura)

03 Pantalla principal (Carrossel):

- Primer Carrossel amb Artistes (Ordenats per nombre d'oreigs).
- Altres Carrossels amb Àlbums (Ordenats per gènere).

04 Pantalla del Buscador:

- Cercador principal on l'usuari pot cercar (Artista, Àlbums, Cançons).

05 Apartat de Filtratge:

- Es pot filtrar per (Gènere, Preu, Format (CD, Vinil, Digital), Àlbums, Artistes, Cançons)

06 Pantalla d'Artistes:

- Si busques un Artista sortirà un icona amb l'Artista.
- Si cliques surten tots els àlbums d'aquest artista.

07 Pantalla d'Àlbums:

- Si busques un àlbum o entres dins dels àlbums d'un artista sortirà l'ícone de l'àlbum.
- Si cliques surten les cançons d'aquest àlbum, nom de l'artista, ícone de l'àlbum i de l'artista i preu.
- Botó afegir a la cistella

08 Pantalla de Cançons/Àlbum:

- Si busques una cançó et sortirà l'àlbum al qual pertany aquesta cançó.
- Si cliques surt el mateix que a l'àlbum.
- A la informació de la cançó sortirà (nom de la cançó, artista, artista secundari (si n'hi ha), durada).

09 Pantalla de Cistella:

- Contindrà els productes que s'han afegit a la cistella (Foto, nom, quantitat, preu total (de quantitat del mateix producte)).
- El mateix amb els altres productes afegits.
- Preu TOTAL de tots els productes.
- Botó Procedir al Pagament.

10 Pantalla de Pagament:

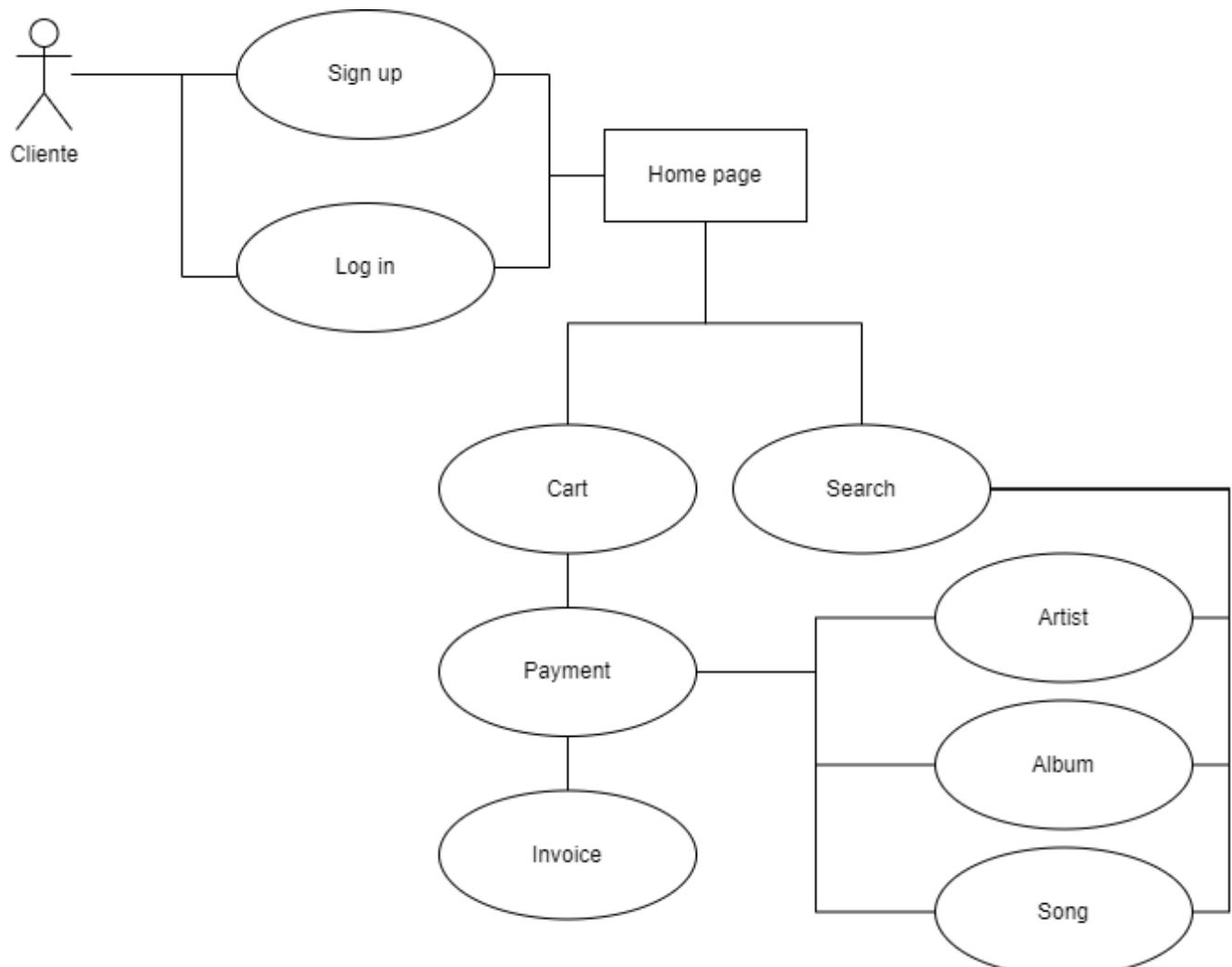
- Un cop es passa a la pantalla de pagament se li demanarà a l'usuari l'adreça de facturació: (País, Província, CP, Telefó, Adreça).
- També se li demanaran les dades bancàries: Número de la targeta (14-19 dígits), Data Caducitat (dia/mes), Codi Seguretat CVV.
- Sortirà el preu TOTAL de tots els productes.
- Botó Acceptar.

11

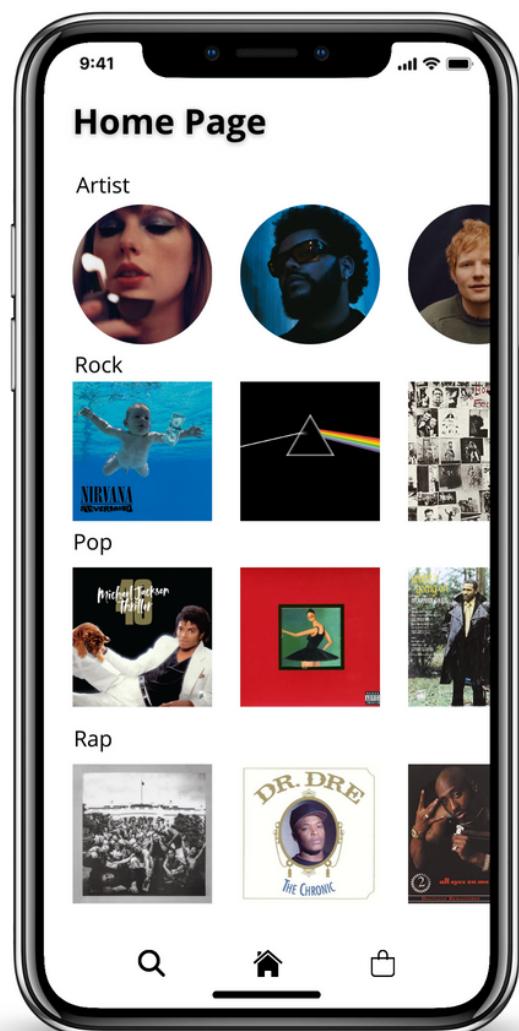
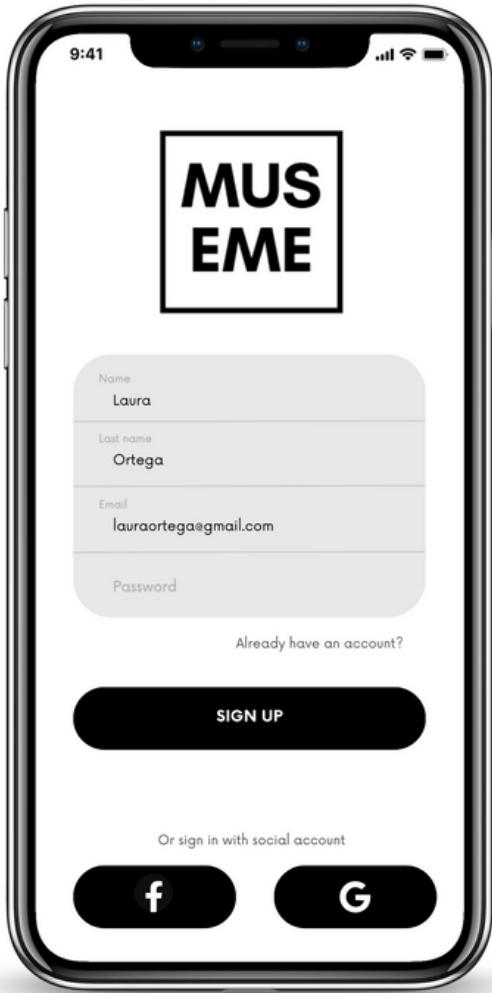
Pantalla de Factura:

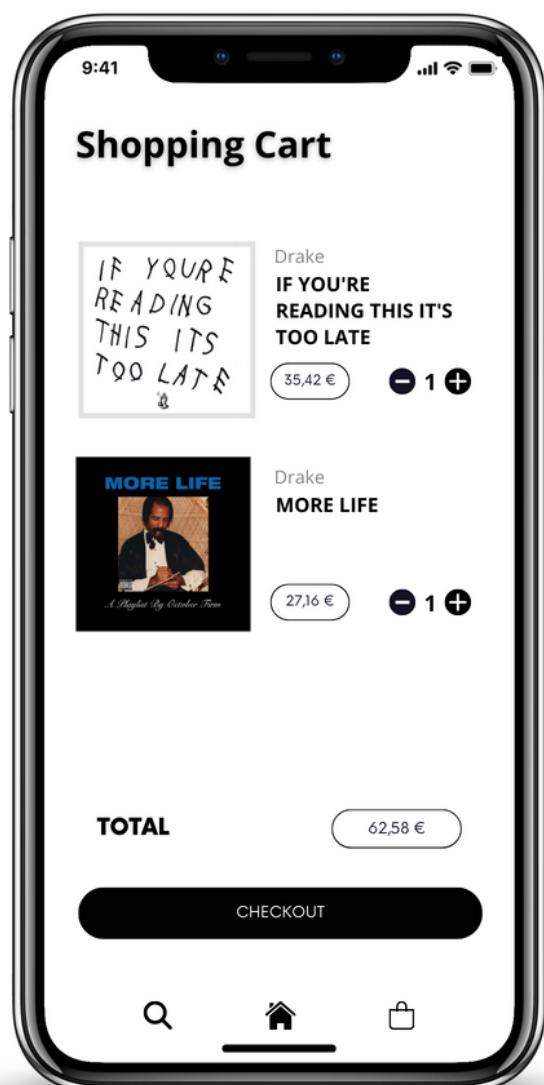
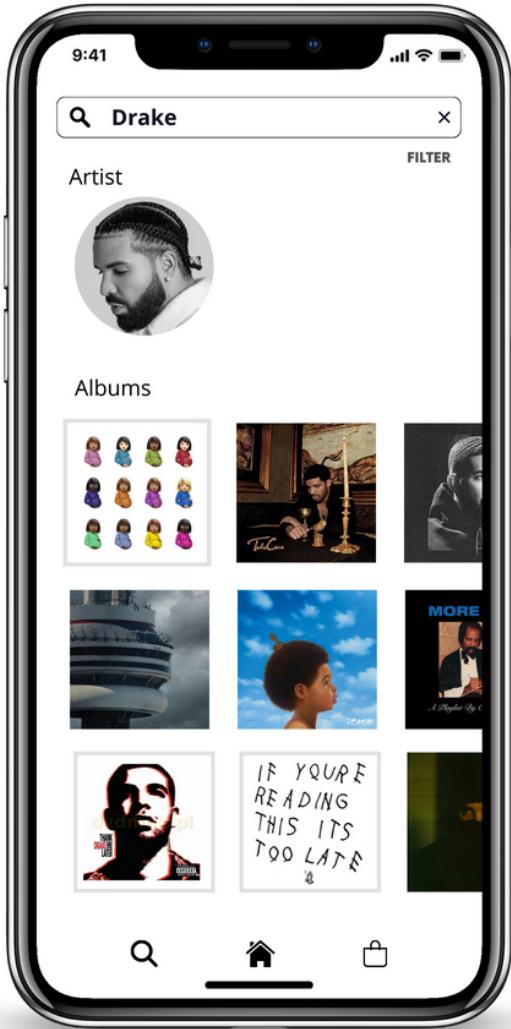
- Nom i cognoms de l'usuari.
- Adreça de facturació.
- Data de la comanda.
- Informació dels productes.
- Total del preu de la comanda.

DIAGRAMA FUNCIONAL USUARI



MOCKUPS





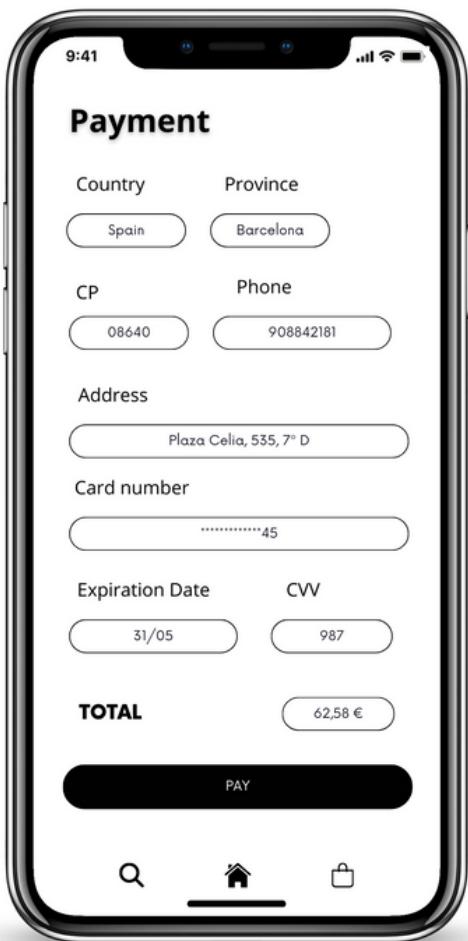
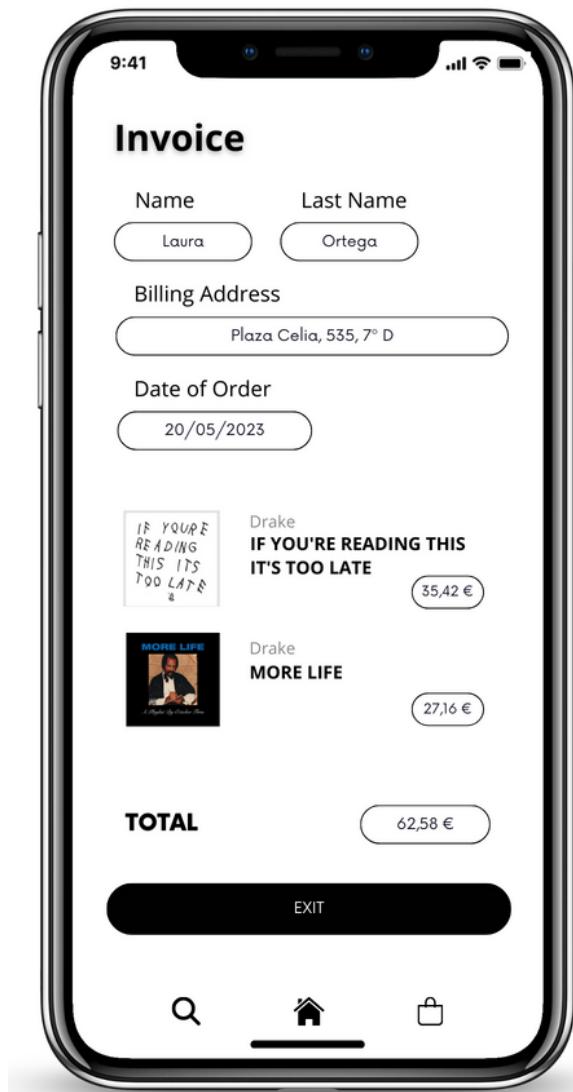
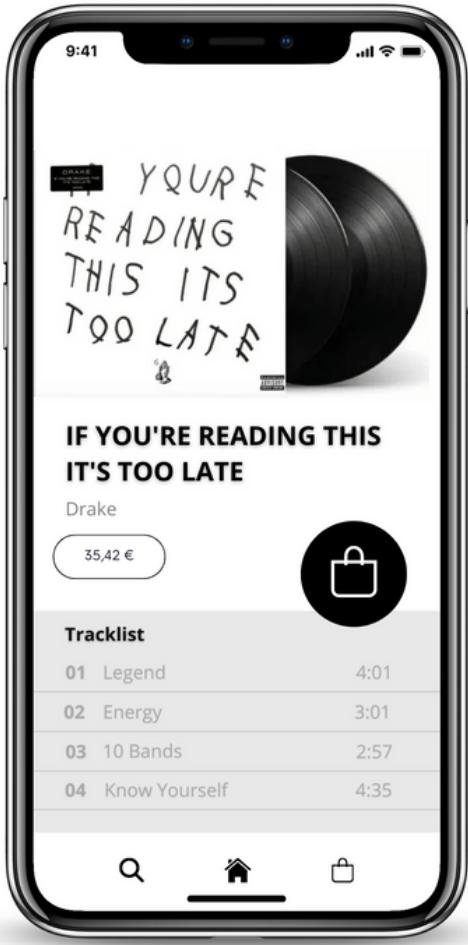


DIAGRAMA ENTITAT-RELACIÓ

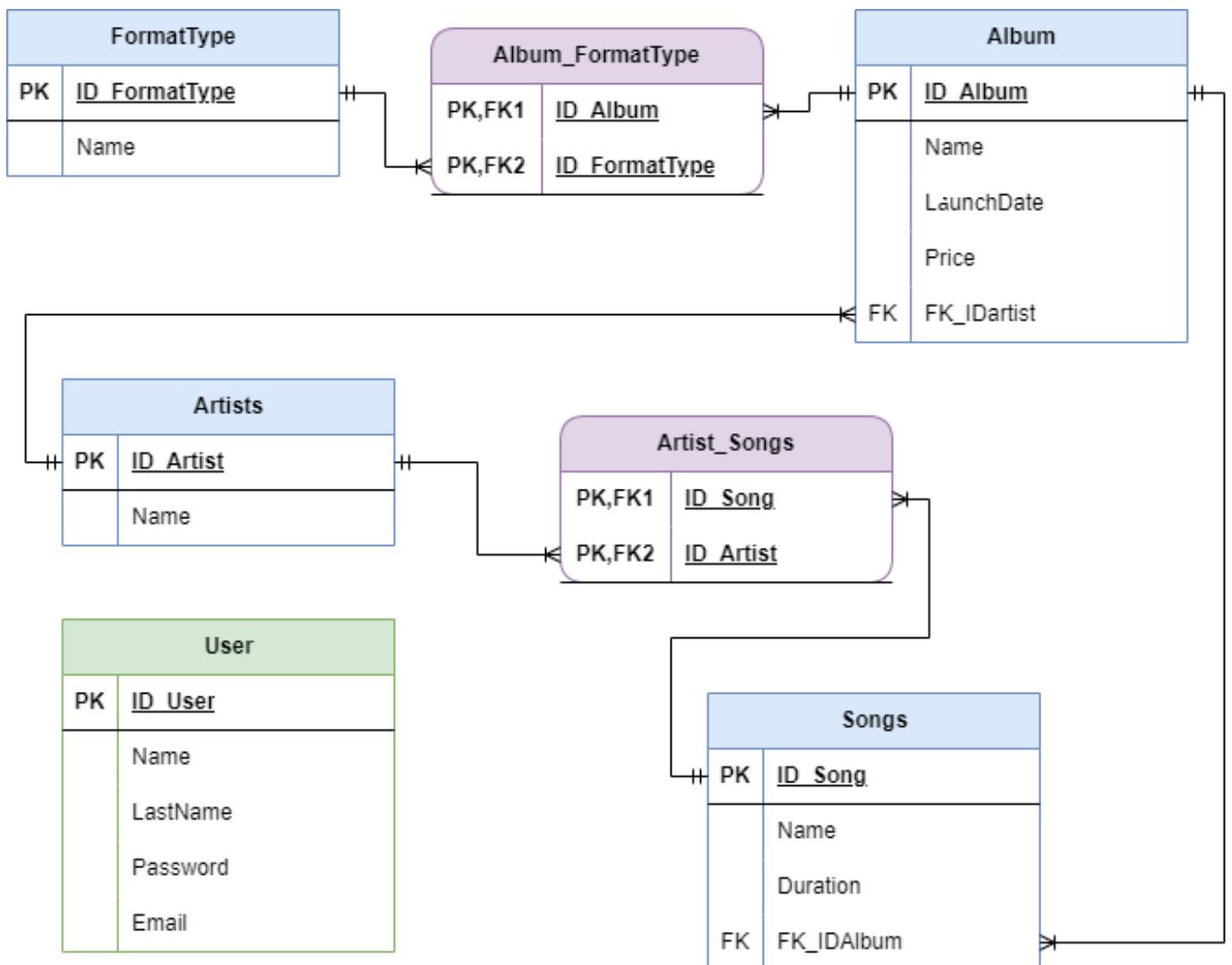


DIAGRAMA UML INICIAL

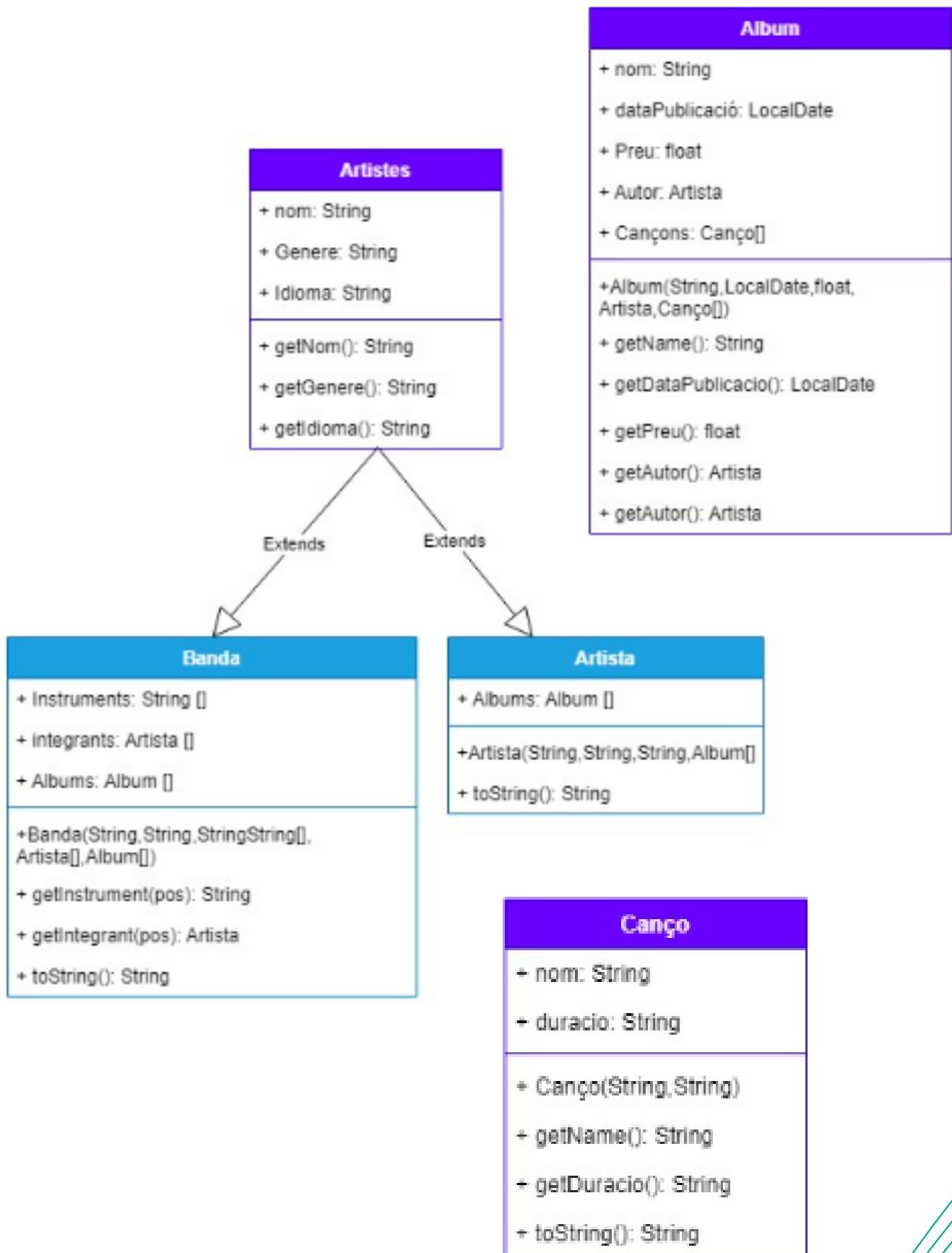


DIAGRAMA UML GENERAT PER L'IDE

<p>albumPageDrakeController</p> <ul style="list-style-type: none"> □ panelFormSignIn: VBox □ banerDrake: ImageView □ btnHonestlyNeverMind: Button □ btnTooLate: Button □ btnNothingWasSame: Button □ btnLog: Button □ btnHome: Button □ btnExit: Button <ul style="list-style-type: none"> ● actionHonestlyNeverMind(event: ActionEvent): void ● actionTooLate(event: ActionEvent): void ● actionNothingWasSame(event: ActionEvent): void ● actionLog(event: ActionEvent): void ● actionHome(event: ActionEvent): void ● actionExit(event: ActionEvent): void 	<p>albumPageMichaelController</p> <ul style="list-style-type: none"> □ panelFormSignIn: VBox □ banerMichael: ImageView □ btnThriller: Button □ btnOffTheWall: Button □ btnBad: Button □ btnLog: Button □ btnHome: Button □ btnExit: Button <ul style="list-style-type: none"> ● actionThriller(event: ActionEvent): void ● actionOffTheWall(event: ActionEvent): void ● actionBad(event: ActionEvent): void ● actionLog(event: ActionEvent): void ● actionHome(event: ActionEvent): void ● actionExit(event: ActionEvent): void
<p>homePageController</p> <ul style="list-style-type: none"> □ panelFormSignIn: VBox □ Drake: Button □ DrakeImagen: ImageView □ Michael: Button □ MichaelImagen: ImageView □ Nirvana: Button □ NirvanaImagen: ImageView □ btnDrake: Button □ btnMichael: Button □ btnNirvana: Button □ btnThriller: Button □ Thriller: ImageView □ btnOffTheWall: Button □ OffTheWall: ImageView □ btnBad: Button □ Bad: ImageView □ btnHonestlyNeverMind: Button □ HonestlyNeverMind: ImageView □ btnTooLate: Button □ Too_Late: ImageView □ btnNothingWasTheSame: Button □ NothingWasSame: ImageView □ btnInUtero: Button □ InUterolImagen: ImageView □ Bleach: Button □ BleachImagen: ImageView □ Nevermind: Button □ NevermindImagen: ImageView □ btnSearch: Button □ btnHome: Button □ btnExit: Button <ul style="list-style-type: none"> ● actionDrake(event: ActionEvent): void ● actionMichael(event: ActionEvent): void ● actionNirvana(event: ActionEvent): void ● actionThriller(event: ActionEvent): void ● actionOffTheWall(event: ActionEvent): void ● actionBad(event: ActionEvent): void ● actionHonestlyNeverMind(event: ActionEvent): void ● actionTooLate(event: ActionEvent): void ● actionNothingWasTheSame(event: ActionEvent): void ● actionInUtero(event: ActionEvent): void ● actionBleach(event: ActionEvent): void ● actionNevermind(event: ActionEvent): void ● actionSearch(event: ActionEvent): void ● actionHome(event: ActionEvent): void ● actionExit(event: ActionEvent): void 	<p>albumPageNirvanaController</p> <ul style="list-style-type: none"> □ panelFormSignIn: VBox □ banerDrake: ImageView □ btnNevermind: Button □ btnInUtero: Button □ btnBleach: Button □ btnLog: Button □ btnHome: Button □ btnExit: Button <ul style="list-style-type: none"> ● actionNevermind(event: ActionEvent): void ● actionInUtero(event: ActionEvent): void ● actionBleach(event: ActionEvent): void ● actionLog(event: ActionEvent): void ● actionHome(event: ActionEvent): void ● actionExit(event: ActionEvent): void
	<p>loginController</p> <ul style="list-style-type: none"> □ panelFormSignIn: VBox □ txtEmail: TextField □ txtPassword: PasswordField □ txtPasswordSignInMask: TextField □ btnRegister: Button □ btnLogin: Button <ul style="list-style-type: none"> ● actionRegister(event: ActionEvent): void ● actionLogin(event: ActionEvent): void
	<p>registerController</p> <ul style="list-style-type: none"> □ panelFormsSignUp: VBox □ txtName: TextField □ txtLastname: TextField □ txtEmail: TextField □ txtPassword: PasswordField □ txtPassword2: PasswordField □ btnRegister: Button <ul style="list-style-type: none"> ● actionRegister(event: ActionEvent): void

C

albumPageDrakeController

- panelFormSignIn: VBox
- banerDrake: ImageView
- btnHonestlyNeverMind: Button
- btnTooLate: Button
- btnNothingWasSame: Button
- btnLog: Button
- btnHome: Button
- btnExit: Button

- actionHonestlyNeverMind(event: ActionEvent): void
- actionTooLate(event: ActionEvent): void
- actionNothingWasSame(event: ActionEvent): void
- actionLog(event: ActionEvent): void
- actionHome(event: ActionEvent): void
- actionExit(event: ActionEvent): void

C

albumPageMichaelController

- panelFormSignIn: VBox
- banerMichael: ImageView
- btnThriller: Button
- btnOffTheWall: Button
- btnBad: Button
- btnLog: Button
- btnHome: Button
- btnExit: Button

- actionThriller(event: ActionEvent): void
- actionOffTheWall(event: ActionEvent): void
- actionBad(event: ActionEvent): void
- actionLog(event: ActionEvent): void
- actionHome(event: ActionEvent): void
- actionExit(event: ActionEvent): void

C

homePageController

- panelFormSignIn: VBox
- Drake: Button
- DrakeImagen: ImageView
- Michael: Button
- MichaelImagen: ImageView
- Nirvana: Button
- NirvanaImagen: ImageView
- btnDrake: Button
- btnMichael: Button
- btnNirvana: Button
- Thriller: ImageView
- OffTheWall: ImageView
- Bad: ImageView
- HonestlyNeverMind: ImageView
- Too_Late: ImageView
- NothingWasSame: ImageView
- inUteroImagen: ImageView
- BleachImagen: ImageView
- NevermindImagen: ImageView
- btnSearch: Button
- btnHome: Button
- btnExit: Button

- actionDrake(event: ActionEvent): void
- actionMichael(event: ActionEvent): void
- actionNirvana(event: ActionEvent): void
- actionSearch(event: ActionEvent): void
- actionHome(event: ActionEvent): void
- actionExit(event: ActionEvent): void

C

loginController

- panelFormSignIn: VBox
- txtEmail: TextField
- txtPassword: PasswordField
- txtPasswordSignInMask: TextField
- btnRegister: Button
- btnLogin: Button

- actionRegister(event: ActionEvent): void
- actionLogin(event: ActionEvent): void

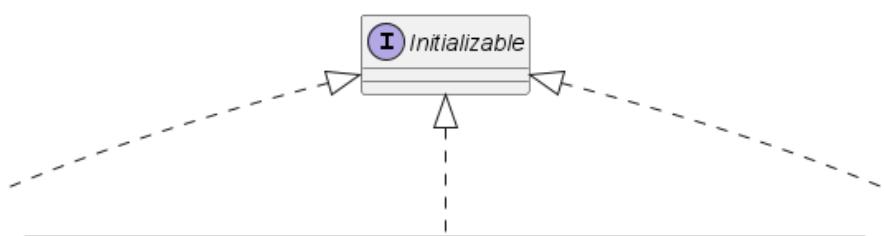
C

albumPageNirvanaController

- panelFormSignIn: VBox
- banerDrake: ImageView
- btnNevermind: Button
- btnInUtero: Button
- btnBleach: Button
- btnLog: Button
- btnHome: Button
- btnExit: Button

- actionNevermind(event: ActionEvent): void
- actionInUtero(event: ActionEvent): void
- actionBleach(event: ActionEvent): void
- actionLog(event: ActionEvent): void
- actionHome(event: ActionEvent): void
- actionExit(event: ActionEvent): void

C	BadController
<ul style="list-style-type: none"> □ panelFormSignIn: VBox □ banerMichael: ImageView □ OffTheWall: ImageView □ Tablageneral: TableView<Map<String, ObservableValue<String>>> □ columName: TableColumn<Map<String, ObservableValue<String>>, String> □ columDuration: TableColumn<Map<String, ObservableValue<String>>, String> □ btnLog: Button □ btnHome: Button □ btnExit: Button 	
<ul style="list-style-type: none"> ● actionLog(event: ActionEvent): void ● actionHome(event: ActionEvent): void ● actionExit(event: ActionEvent): void ● initialize(arg0: URL, arg1: ResourceBundle): void 	

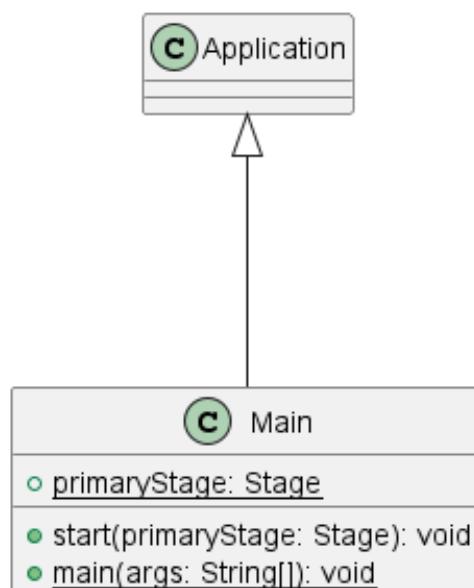


C	BleachController
<ul style="list-style-type: none"> □ panelFormSignIn: VBox □ banerDrake: ImageView □ Tablageneral: TableView<Map<String, ObservableValue<String>>> □ columName: TableColumn<Map<String, ObservableValue<String>>, String> □ columDuration: TableColumn<Map<String, ObservableValue<String>>, String> □ btnLog: Button □ btnHome: Button □ btnExit: Button 	
<ul style="list-style-type: none"> ● actionLog(event: ActionEvent): void ● actionHome(event: ActionEvent): void ● actionExit(event: ActionEvent): void ● initialize(arg0: URL, arg1: ResourceBundle): void 	

C	HonestlyNeverMindController
<ul style="list-style-type: none"> □ panelFormSignIn: VBox □ banerDrake: ImageView □ btnHonestlyNeverMind: Button □ HonestlyNeverMind: ImageView □ Tablageneral: TableView<Map<String, ObservableValue<String>>> □ columName: TableColumn<Map<String, ObservableValue<String>>, String> □ columDuration: TableColumn<Map<String, ObservableValue<String>>, String> □ btnLog: Button □ btnHome: Button □ btnExit: Button 	
<ul style="list-style-type: none"> ● actionHonestlyNeverMind(event: ActionEvent): void ● actionLog(event: ActionEvent): void ● actionHome(event: ActionEvent): void ● actionExit(event: ActionEvent): void ● initialize(arg0: URL, arg1: ResourceBundle): void 	

C	registerController
□	panelFormsSignUp: VBox
□	txtName: TextField
□	txtLastname: TextField
□	txtEmail: TextField
□	txtPassword: PasswordField
□	txtPassword2: PasswordField
□	btnRegister: Button
●	actionRegister(event: ActionEvent): void

C	MyMethods
●	<u>CheckName(nombre: String, apellido: String): boolean</u>
●	<u>CheckEmail(email: String): boolean</u>
●	<u>CheckPassword(password: String): boolean</u>
●	<u>CheckEqualPassword(password: String, password2: String): boolean</u>
●	<u>AlreadyRegister(email: String): boolean</u>
●	<u>CheckLogin(password: String, email: String): boolean</u>
●	<u>NotAlreadyRegister(email: String): boolean</u>
●	<u>fillTable(Album: String, Tablageneral: TableView<Map<String, ObservableValue<String>>): void</u>



KANBANFLOW

Done		+
Today		
Creació base de dades	ESM	
Connexió base de dades amb eclipse	ESM	
JavaFX Views	MM	
Creació de les classes i metodes	ESM	
Monday, 22 May		
Model ER i Relacional de la BBDD	MM	
Thursday, 18 May		
Diagrames UML	ESM	
Anàlisi funcional	MM	
Mockups	MM	