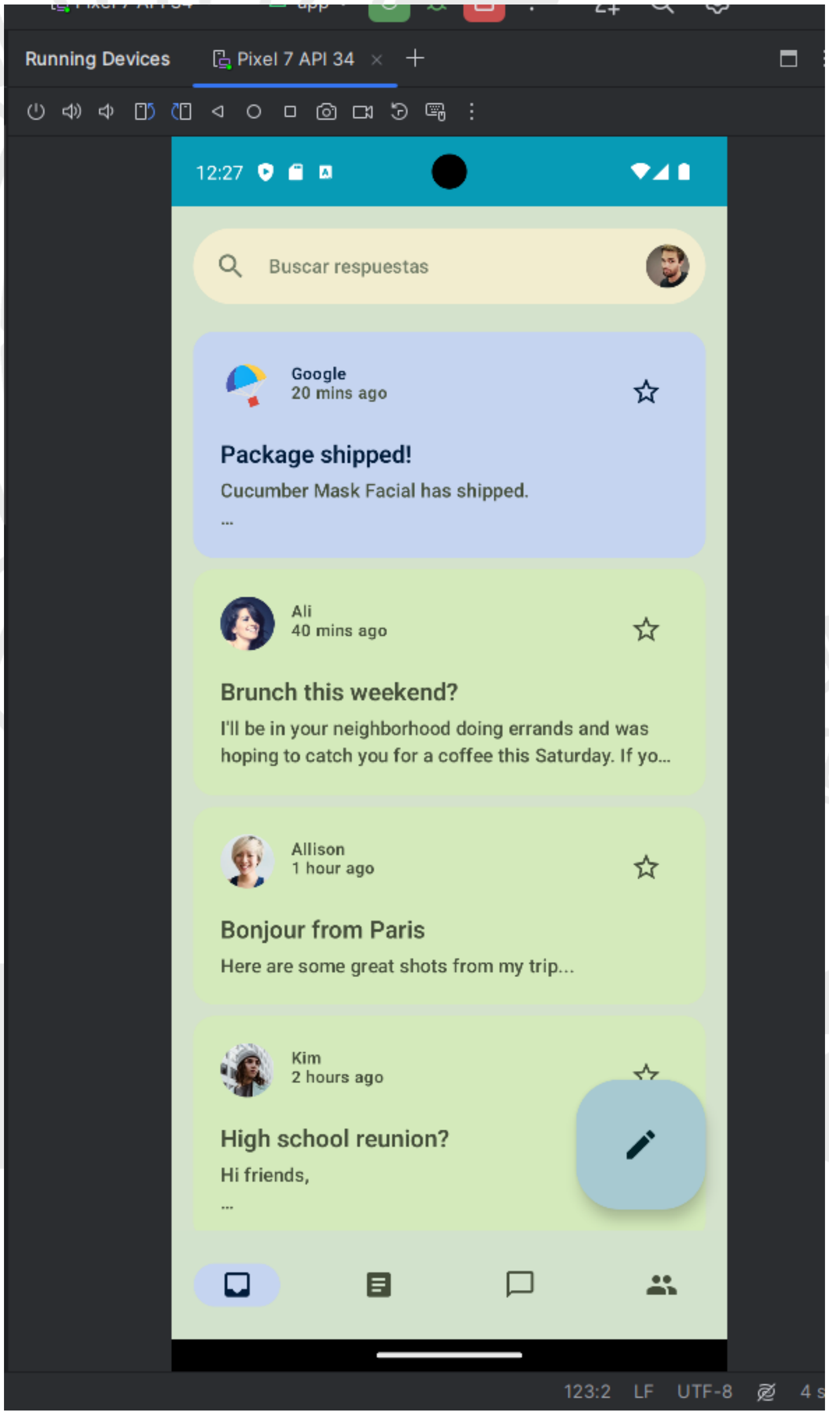
PROGRAMACIÓN MÓVIL || COMPRENSIÓN: IMPLEMENTAR MATERIAL DESIGN 3

UA: Programación Móvil POR: MARTÍNEZ PANTALEÓN JOSÉ DE JESÚS

Fecha: 09 de junio de 2024 Secuencia: 6 N M 6 1

PRUEBAS DEL SISTEMA DE Q&A:

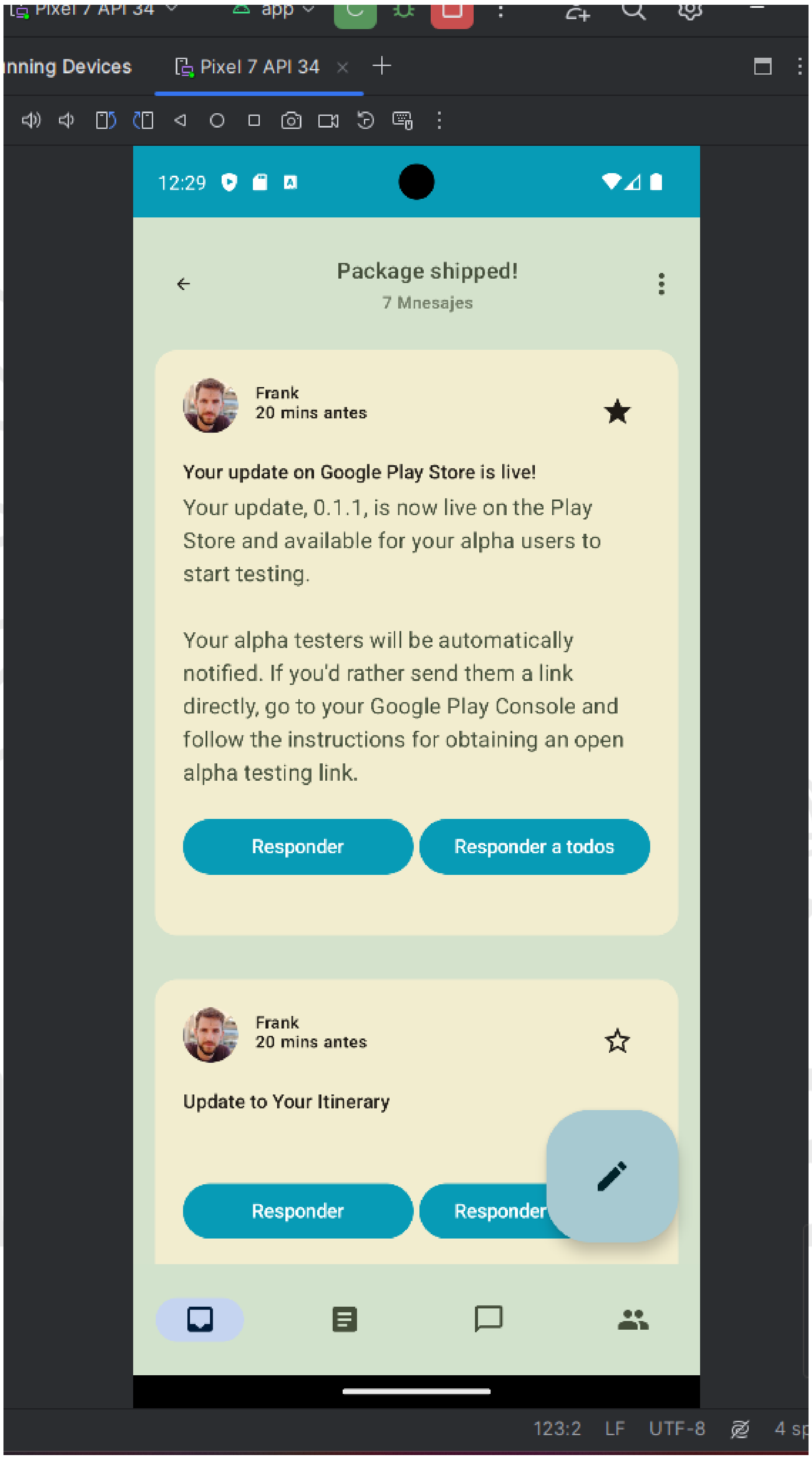


Google
20 mins ago
Package shipped!
Cucumber Mask Facial has shipped.
...

Ali
40 mins ago
Brunch this weekend?
I'll be in your neighborhood doing errands and was hoping to catch you for a coffee this Saturday. If yo...

Allison
1 hour ago
Bonjour from Paris
Here are some great shots from my trip...

Kim
2 hours ago
High school reunion?
Hi friends,
...



Package shipped!

7 Mnesajes

Frank
20 mins antes

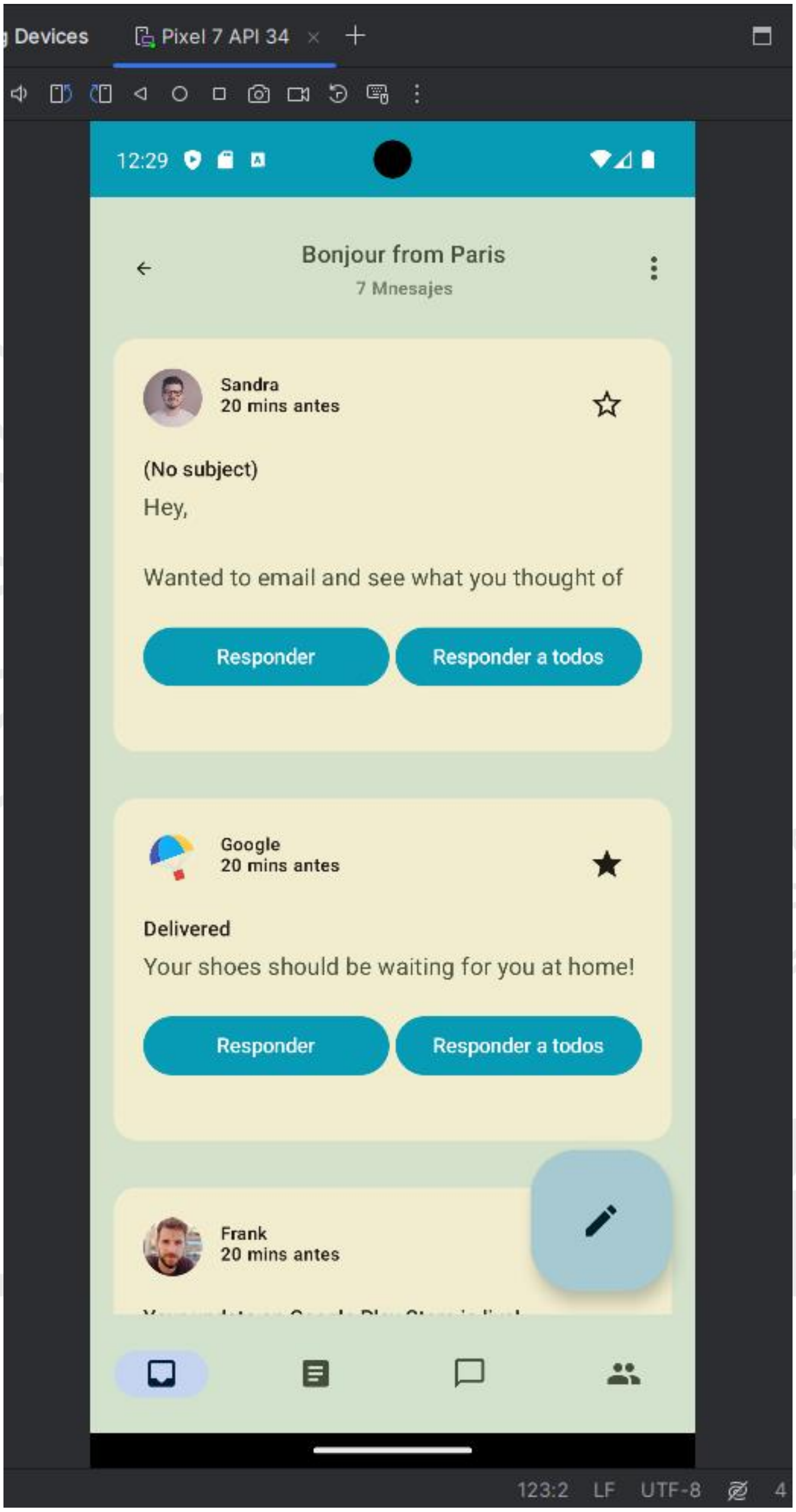
Your update on Google Play Store is live!

Your update, 0.1.1, is now live on the Play Store and available for your alpha users to start testing.

Your alpha testers will be automatically notified. If you'd rather send them a link directly, go to your Google Play Console and follow the instructions for obtaining an open alpha testing link.

Frank
20 mins antes

Update to Your Itinerary



Devices Pixel 7 API 34

12:29

Bonjour from Paris

7 Mnesajes

Sandra
20 mins antes

(No subject)

Hey,

Wanted to email and see what you thought of

Google
20 mins antes

Delivered

Your shoes should be waiting for you at home!

Frank
20 mins antes

Modificación del esquema de colores

Código:
```
package com.example.compose

import androidx.compose.ui.graphics.Color

// Definiciones de colores de la nueva paleta
val md_theme_light_primary = Color(0xFF079BB6)
val md_theme_light_onPrimary = Color(0xFFFFFFFF)
val md_theme_light_primaryContainer = Color(0xFF9CD2D3)
val md_theme_light_onPrimaryContainer = Color(0xFF002026)
val md_theme_light_secondary = Color(0xFF4A6EB0)
val md_theme_light_onSecondary = Color(0xFFFFFFFF)
val md_theme_light_secondaryContainer = Color(0xFFC5D4F0)
val md_theme_light_onSecondaryContainer = Color(0xFF001B3A)
val md_theme_light_tertiary = Color(0xFF114E5F)
val md_theme_light_onTertiary = Color(0xFFFFFFFF)
val md_theme_light_tertiaryContainer = Color(0xFFA7C9D1)
val md_theme_light_onTertiaryContainer = Color(0xFF00232A)
val md_theme_light_error = Color(0xFFBA1A1A)
val md_theme_light_errorContainer = Color(0xFFFFDAD6)
val md_theme_light_onError = Color(0xFFFFFFFF)
val md_theme_light_onErrorContainer = Color(0xFF410002)
val md_theme_light_background = Color(0xFFF2EDCF)
val md_theme_light_onBackground = Color(0xFF1F1B16)
val md_theme_light_surface = Color(0xFFF2EDCF)
val md_theme_light_onSurface = Color(0xFF1F1B16)
val md_theme_light_surfaceVariant = Color(0xFFD4EABB)
val md_theme_light_onSurfaceVariant = Color(0xFF444E3A)
val md_theme_light_outline = Color(0xFF737E69)
val md_theme_light_inverseOnSurface = Color(0xFFF2EDCF)
val md_theme_light_inverseSurface = Color(0xFF34302A)
val md_theme_light_inversePrimary = Color(0xFF79C3C7)
val md_theme_light_shadow = Color(0xFF000000)
val md_theme_light_surfaceTint = Color(0xFF079BB6)
val md_theme_light_outlineVariant = Color(0xFFC5D4F0)
val md_theme_light_scrim = Color(0xFF000000)

val md_theme_dark_primary = Color(0xFF79C3C7)
val md_theme_dark_onPrimary = Color(0xFF00353B)
val md_theme_dark_primaryContainer = Color(0xFF004F57)
val md_theme_dark_onPrimaryContainer = Color(0xFF9CD2D3)
val md_theme_dark_secondary = Color(0xFFA4B6E0)
val md_theme_dark_onSecondary = Color(0xFF002D55)
val md_theme_dark_secondaryContainer = Color(0xFF1E427B)
val md_theme_dark_onSecondaryContainer = Color(0xFFC5D4F0)
val md_theme_dark_tertiary = Color(0xFF6FA2A9)
val md_theme_dark_onTertiary = Color(0xFF00363E)
val md_theme_dark_tertiaryContainer = Color(0xFF004D5A)
val md_theme_dark_onTertiaryContainer = Color(0xFFA7C9D1)
val md_theme_dark_error = Color(0xFFFFB4AB)
val md_theme_dark_errorContainer = Color(0xFF93000A)
val md_theme_dark_onError = Color(0xFF690005)
val md_theme_dark_onErrorContainer = Color(0xFFFFDAD6)
val md_theme_dark_background = Color(0xFF1F1B16)
val md_theme_dark_onBackground = Color(0xFFEAE1D9)
val md_theme_dark_surface = Color(0xFF1F1B16)
val md_theme_dark_onSurface = Color(0xFFEAE1D9)
val md_theme_dark_surfaceVariant = Color(0xFF444E3A)
val md_theme_dark_onSurfaceVariant = Color(0xFFC5D4F0)
val md_theme_dark_outline = Color(0xFF8A927F)
val md_theme_dark_inverseOnSurface = Color(0xFF1F1B16)
val md_theme_dark_inverseSurface = Color(0xFFEAE1D9)
val md_theme_dark_inversePrimary = Color(0xFF079BB6)
val md_theme_dark_shadow = Color(0xFF000000)
val md_theme_dark_surfaceTint = Color(0xFF79C3C7)
val md_theme_dark_outlineVariant = Color(0xFF444E3A)
val md_theme_dark_scrim = Color(0xFF000000)
```

Cambio del tema

```kotlin
package com.example.reply.ui.theme

import android.app.Activity
import android.os.Build
import androidx.compose.foundation.isSystemInDarkTheme
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.darkColorScheme
import androidx.compose.material3.lightColorScheme
import androidx.compose.runtime.Composable
import androidx.compose.runtime.SideEffect
import androidx.compose.ui.graphics.toArgb
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.platform.LocalView
import androidx.core.view.WindowCompat
import com.example.compose.*

private val LightColors = lightColorScheme(
    primary = md_theme_light_primary,
    onPrimary = md_theme_light_onPrimary,
    primaryContainer = md_theme_light_primaryContainer,
    onPrimaryContainer = md_theme_light_onPrimaryContainer,
    secondary = md_theme_light_secondary,
    onSecondary = md_theme_light_onSecondary,
    secondaryContainer = md_theme_light_secondaryContainer,
    onSecondaryContainer = md_theme_light_onSecondaryContainer,
    tertiary = md_theme_light_tertiary,
    onTertiary = md_theme_light_onTertiary,
    tertiaryContainer = md_theme_light_tertiaryContainer,
    onTertiaryContainer = md_theme_light_onTertiaryContainer,
    error = md_theme_light_error,
    errorContainer = md_theme_light_errorContainer,
    onError = md_theme_light_onError,
    onErrorContainer = md_theme_light_onErrorContainer,
    background = md_theme_light_background,
    onBackground = md_theme_light_onBackground,
    surface = md_theme_light_surface,
    onSurface = md_theme_light_onSurface,
    surfaceVariant = md_theme_light_surfaceVariant,
    onSurfaceVariant = md_theme_light_onSurfaceVariant,
    outline = md_theme_light_outline,
    inverseOnSurface = md_theme_light_inverseOnSurface,
    inverseSurface = md_theme_light_inverseSurface,
    inversePrimary = md_theme_light_inversePrimary,
    surfaceTint = md_theme_light_surfaceTint,
    outlineVariant = md_theme_light_outlineVariant,
    scrim = md_theme_light_scrim,
)

private val DarkColors = darkColorScheme(
    primary = md_theme_dark_primary,
    onPrimary = md_theme_dark_onPrimary,
    primaryContainer = md_theme_dark_primaryContainer,
    onPrimaryContainer = md_theme_dark_onPrimaryContainer,
    secondary = md_theme_dark_secondary,
    onSecondary = md_theme_dark_onSecondary,
    secondaryContainer = md_theme_dark_secondaryContainer,
    onSecondaryContainer = md_theme_dark_onSecondaryContainer,
    tertiary = md_theme_dark_tertiary,
    onTertiary = md_theme_dark_onTertiary,
    tertiaryContainer = md_theme_dark_tertiaryContainer,
    onTertiaryContainer = md_theme_dark_onTertiaryContainer,
    error = md_theme_dark_error,
    errorContainer = md_theme_dark_errorContainer,
    onError = md_theme_dark_onError,
    onErrorContainer = md_theme_dark_onErrorContainer,
    background = md_theme_dark_background,
    onBackground = md_theme_dark_onBackground,
    surface = md_theme_dark_surface,
    onSurface = md_theme_dark_onSurface,
    surfaceVariant = md_theme_dark_surfaceVariant,
    onSurfaceVariant = md_theme_dark_onSurfaceVariant,
    outline = md_theme_dark_outline,
    inverseOnSurface = md_theme_dark_inverseOnSurface,
    inverseSurface = md_theme_dark_inverseSurface,
    inversePrimary = md_theme_dark_inversePrimary,
    surfaceTint = md_theme_dark_surfaceTint,
    outlineVariant = md_theme_dark_outlineVariant,
    scrim = md_theme_dark_scrim,
)

@Composable
fun AppTheme(
    useDarkTheme: Boolean = isSystemInDarkTheme(),
    content: @Composable () -> Unit
) {
    val colors = if (useDarkTheme) {
        DarkColors
    } else {
        LightColors
    }

    val view = LocalView.current
    if (!view.isInEditMode) {
        SideEffect {
            val window = (view.context as Activity).window
            window.statusBarColor = colors.primary.toArgb()
            WindowCompat.getInsetsController(window,
view).isAppearanceLightStatusBars = useDarkTheme
        }
    }

    MaterialTheme(
        colorScheme = colors,
        typography = typography,
        shapes = shapes,
        content = content
    )
}
```